Railway scheduling in the presence of uncertainties

Nattapol Paisarnvirosrak

Centre for Operational Research and Logistics (CORL)

School of Mathematics and Physics

University of Portsmouth

The thesis is submitted for the degree of

Doctor of Philosophy

September 2019

Declaration

Whilst registered as a candidate for the degree of Doctor of Philosophy, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award. This thesis comprises 38,560 words.

Nattapol Paisarnvirosrak

September 2019

Acknowledgements

First, I would like to thank my supervisors, Prof Djamila Ouelhadj and Dr Banafsheh Khosravi for their help, support, patience and motivation during the research and development of this PhD thesis. I also wish to express my sincere appreciation to the University of Portsmouth Doctoral Programme for providing the optimal learning environment. A big thank you goes to the numerous staff at the Department of Mathematics and Physics and the Graduate School for their valuable guidance and support.

My family steadfastly supported me both emotionally and spiritually throughout this PhD journey along the road of life while my friends Hassana, Mohanad, Abdulwahab and all the Thai PhD students at the School of Mathematics and Physics freely shared and exchanged ideas throughout countless fruitful academic discussions. Staff at the State Railway of Thailand offered technical advice and provided data for the Thai rail network which were used in the computational experiments.

Finally, I am also grateful to my sister Sasikarn, my friend Phornprom, Kullachet and all the Thai PhD students at the University of Portsmouth for their unwavering help, support and encouragement. Guys, you made my time as a PhD student really enjoyable!

Abstract

Scheduling and rescheduling play a central role in day-to-day railway operations. Trains in a railway network are scheduled and controlled according to a timetable. However, the proposed timetable cannot be always followed because of unpredictable disruptions caused by many factors including excessive dwell times at stations, infrastructure and/or train faults and late arrival of crew. When trains do not operate according to the schedule, even by only a few seconds, there is an increased likelihood that they will cause conflict with other trains, resulting in further delays. This issue is even more critical in a congested network with high interconnection between trains because delays are easily propagated across the whole network, affecting all interconnected trains in the network.

The thesis focuses on railway scheduling problems (RSPs) by addressing the deterministic RSP and the stochastic RSP. Novel models and solution methods are proposed to solve these problems. We developed an optimisation model based on a set partitioning model with the main objective to minimise the total delay of trains while considering passenger safety and regulation principles including running times, headway and signalling system constraints. Moreover, we proposed heuristic and metaheuristic methods enhanced by biased randomisation and local search techniques to solve the deterministic RSP. These included Iterated Greedy with Biased Randomised (IG-BR), Biased Randomised Iterated Greedy with Local Search (BR-IG-LS) and Biased Randomised Variable Neighbourhood Search (BR-VNS).

Furthermore, we developed a stochastic optimisation model to handle delays and minimise the total delay of trains. The heuristics and metaheuristics methods which combined with Monte Carlo Simulation (MCS) to generate stochastic random delays including Sim-Iterated Greedy with Biased Randomised (S-IG-BR), Sim-Biased Randomised Iterated Greedy with Local Search (S-BR-IG-LS) and Sim-Biased Randomised Variable Neighbourhood Search (S- BR-VNS).

To the best of our knowledge, this is the first time that these proposed methods have been used to solve deterministic and stochastic RSPs.

To evaluate the performance of the proposed optimisation models and the solution methods, we conducted computational experiments using real-world case studies from the Southeastern train operating company, UK and State Railway of Thailand. Results

indicated that the proposed methods outperformed the solutions adopted by the railway companies and/or existing mixed integer linear programming (MILP) obtained by CPLEX optimisation software package.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Glossary of Abbreviations

BDS-Det = Best Deterministic Solution in Deterministic scenario

BDS-Stoch = Best Deterministic Solution in Stochastic scenario

BR-IG-LS = Biased Randomised Iterated Greedy Local Search

BR-VNS = Biased Randomised Variable Neighbourhood Search

CPP = Crew Pairing Problem

COPs = Combinatorial Optimisation Problems

DP = Dynamic Programming

GA = Genetic Algorithm

IG = Iterated Greedy

IG-BR = Iterated Greedy with Biased Randomised

ILS = Iterative Local Search

IP = Integer Programming

LP = Linear programming

LS = Local Search

MCS = Monte Carlo Simulation

MILP = Mixed Integer Linear Programming

MIP = Mixed Integer Programming

OR = Operational Research

RD = Relative Deviation

RSP = Railway Scheduling Problem

S-BR-IG-LS = Sim Biased Randomised Iterated Greedy Local Search

S-BR-VNS = Sim Biased Randomised Variable Neighbourhood Search

S-IG-BR = Sim Iterated Greedy with Biased Randomised

S-VNS = Sim Variable Neighbourhood Search Algorithm

SA = Simulated Annealing

SB = Shifting Bottleneck

SSA = Sequential Search Algorithm

SSDP = Southeastern Service Disruption Procedure

TS = Tabu Search

VNS = Variable Neighbourhood Search

VRP = Vehicle Routing Problem

Chapter 1: Introduction

1.1. Background and Motivation

Scheduling and rescheduling problems play a central role in day-to-day railway operations. Trains in a railway network are scheduled and controlled according to a timetable. Schedules are designed to be conflict free, that is, they should not contain any situations where a train is restricted in its scheduled movement by other trains.

However, in practice not all trains run according to the schedule due to delays such as excessive dwell times at stations, longer running time, infrastructure and/or train faults and late arrival of crew. When trains do not operate according to the schedule, even by only a few seconds, there is an increased likelihood that they will cause conflict with other trains which will then also be delayed. This issue is more serious in a congested network with high interconnections between trains because delays will quickly propagate to the whole network and affect all trains in the system.

Many countries in the world invest in technology as systems to improve reliability, capacity utilisation and effective response to optimise railway management which including train control and train traffic management systems. Moreover, the easiest and cheapest way to improve the rail network is an effective scheduling and rescheduling process since costs of constructing new infrastructures and maintenance of rolling stock are prohibitive. Therefore, utilisation of railway systems can be improved by creating better operating plans or effective rescheduling processes.

Over the last few decades, railway traffic has expanded considerably to satisfy the increase in customer demand. This has created the need to optimise the use of railway infrastructures and improve the efficiency of railway management. The real-world rail network is large and complex. Railway companies cannot attain the full capacity limit of their infrastructures; therefore, huge opportunities exist to improve the railway process.

The most important process in real-time operation is rescheduling to provide immediate decisions to respond to unexpected events. Railway scheduling problem (RSP) aim to decrease total train delay during execution and thereby increase customer satisfaction. In other words, they should be able to regenerate a new schedule within a time limit which is the focus of this thesis.

According to the literature review in Chapter 2, many OR methods have been used to solve RSPs including exact algorithms, heuristics and metaheuristics. These techniques include branch and bound (D'Ariano et al., 2007), local search (Brucker et al., 2005), Shifting Bottleneck (Khosravi et al., 2012), Iterative Local Search (Corman, Ariano, Marra, Pacciarelli, & Samà, 2017), Sequential Search Algorithm (Tian & Niu, 2019), Genetic Algorithms (Ping et al., 2001; Nitisiri et al. 2019), Tabu Search (Törnquist et al., 2005; Samà et al., 2017), Simulated Annealing (Törnquist et al., 2005), and Variable Neighbourhood Search (Samà et al., 2017).

A detailed review of the related literature showed that the Iterated Greedy (IG) algorithm has not been used to solve the deterministic RSP but provided an effective solution in similar optimisation problems. Moreover, only one paper used Variable Neighbourhood Search (VNS) to solve the deterministic RSP (Samà et al., 2017). However, an information gap exists in the exploration of the combination of IG with Biased Randomised (IG-BR), Biased Randomised IG with Local Search (BR-IG-LS) and Biased Randomised VNS (BR-VNS) approaches for both deterministic and stochastic RSPs. A biased randomised heuristic allows the generation of good quality solutions using a skewed probability distribution to guide the solution construction process. The local search heuristic allows an increase in the solution search space to improve solution quality and reduce computation time.

Recently, studies on stochastic optimisation problems have used a combination of simulation and optimisation approaches to solve complex real-life problems and also helped to deal with more realistic and complex scenarios. The Sim-heuristics or sim-metaheuristics approach is a particular case of simulation-based optimisation which combines a heuristic or metaheuristic algorithm with simulation approaches. The potential of simulation techniques has been widely proven, especially with regard to stochastic behaviour in a real system, previously addressed using simulation (Glover et al., 1996). A stochastic system is a set of dynamic interdependent components where some variable values change randomly. Therefore, simulation processes with stochastic variables are related to the basic mechanism (Juan et al., 2015). Examples of the application of simulation-based optimisation can be found in scheduling mechanism manufacturing (Chong et al., 2003; Frantzén et al., 2011), flow shop scheduling (Yang et al., 2004), job shop scheduling (Arakawa et al., 2003; Klemmt et al., 2009; Nicoară et al., 2011;

Korytkowski et al., 2013), train timetabling problem (Sajedinejad et al., 2011), train transit problem (Hassannayebi et al., 2014).

To the best of our knowledge, literature review contained no paper proposing the use of sim-heuristics or sim-metaheuristics to solve the stochastic RSP problem. This thesis, therefore, is the first to apply Sim-Iterated Greedy with Biased Randomised (S-IG-BR), Sim-Biased Randomised Iterated Greedy with Local Search (S-BR-IG-LS) and Sim-Biased Randomised Variable Neighbourhood Search (S-BR-VNS) method to the stochastic RSP problem to reduce total train delay. Solutions from the stochastic RSP were then compared to the deterministic RSP. The results showed that our simulation-based optimisation was more effective when compared to the deterministic RSP. Two experimental case studies were considered as (i) Southeastern, a train company operating passenger rail services in the London Bridge area, and Kent, South East of the UK by Khosravi (2013) and (ii) The State Railway of Thailand (2019), including four main lines as Northern line, Eastern line, North-eastern line and Southern line.

In the UK, one of the main transportation modes is the railway as one of the best regions for railway operation. Moreover, there are several train operating companies (TOC's) in the UK who are operated the passenger train on the same infrastructure which provided by the Nation Rail, so all TOC's work together and shared their information under the supervision of the Network Rail. Between 4 February 2018 and 2 February 2019 only 62.5% of trains arrived on time (Network Rail, 2019). Therefore, there is a huge opportunity to improve the performance of the UK rail network. In Thailand, most of the rail system in operation is old-fashioned and dates from the last century. Performance is very low and not up-to-date. Here, we adapted the UK railway system to improve the network in Thailand. Moreover, the Thai Government predicts double the numbers of passengers by 2027 and plans to build new infrastructures to serve 61 provinces (Jotikasthira, 2018). This requires a new management system.

Therefore, new techniques were employed as operational research for scheduling and rescheduling to increase the performance of railway operations and customer satisfaction for both the UK and Thai rail networks.

1.2. Aims and Objectives

This research focused on deterministic and stochastic RSPs to reduce total train delays of railway networks. The RSP deals with the uncertainty of delays that occur in the railway process.

For the RSP with deterministic delay, we used sets of instances or historical data. All delays were known in advance before we started to solve a problem.

For the RSP with stochastic delay, we did not know the exact delay before the unexpected event occurred; therefore, we used a simulation model to generate the delay to the railway network by using probability distribution.

This research developed optimisation models and used proposed solution methods which have been successful in other optimisation problems to solve both the deterministic and stochastic RSP.

- For the deterministic RSP, this research aimed to implement an optimisation model which adapted from Khosravi et al. (2012) and algorithm which provided a better solution than currently used by the railway company.
- For the stochastic RSP, we used sim-optimisation to simulation a real situation. The delay was randomly presented in the network to solve RSP and measure the quality and reliability of the solution methods.

The overall aim of this thesis was to develop optimisation models and solution methods to efficiently solve both deterministic and stochastic RSPs as follows:

- Conduct a literature review on deterministic and stochastic RSPs including railway planning processes, topology, signalling systems, solution methods and how to deal with disturbances and/or disruptions.
- Formulate an optimisation model to solve the RSP by considering the characteristics of the UK and Thai railway networks. The objective function of this model was to minimise the total delay of all trains with consideration for train priority. In this research, we considered two benchmark problems collected from:
  - ➢ Southeastern train company, United Kingdom - the London Bridge area and Kent, South East of the UK by Khosravi (2013).
  - ➢ State Railway of Thailand (2019) - the Thai rail network including four main lines as Northern line, Eastern line, North-eastern line and Southern line.

- Implement an Iterated Greedy (IG) algorithm with biased randomisation, Biased Randomised IG algorithm with local search and Variable Neighbourhood Search (VNS) and Biased Randomised VNS to solve the deterministic RSP.
- Formulate a stochastic optimisation model to solve the stochastic RSP by considering the characteristics of the UK and Thai railway networks.
- Implement a Sim-IG algorithm with biased randomisation, Sim-Biased Randomised IG algorithm with local search and Sim-VNS and Sim-Biased Randomised VNS to solve the stochastic RSP. In addition, we selected Monte Carlo Simulation (MCS) to combine the IG algorithm with biased randomisation, Biased Randomised IG algorithm with local search and VNS and Biased Randomised VNS.
- Evaluate the performance of optimisation models and algorithms by comparing experimental results with CPLEX commercial software on the UK case study and historical data from the company on the Thai case study. The reason of using CPLEX as one of our results because the fact that all case studies have not been used before in the literature. Therefore, the CPLEX commercial software, which provided the optimal solution, used to compare and evaluate the performance of the proposed solution methods.
- Conduct experiments to regenerate a new schedule for controlling the railway network.

1.3. Contributions

This thesis developed deterministic and stochastic optimisation models and new solution methods to solve the RSP using two types of signalling systems based on characteristics of the benchmark problems as four-aspect signal (UK-RSP) and two-aspect signal (Thai-RSP). To achieve the objectives considered in section 1.2, a summary of the main contributions of this thesis is provided below; full details will be explained in the study.

A. Optimisation model for the deterministic UK-RSP and Thai-RSP

An optimisation model which adapted from Khosravi et al. (2012) was proposed for solving the UK-RSP and the Thai-RSP using deterministic delays compounded in each set of instances. The objective is to minimise the total delay of all trains, considered two types of signalling constraint.

B. Solution methods for the deterministic UK-RSP and Thai-RSP

- Iterated Greedy with Biased Randomised (IG-BR) and Biased Randomised Iterated Greedy with local search (BR-IG-LS)

With regard to the literature, some studies indicated that the IG algorithm provided an effective solution in similar optimisation problems such as Task assignment (Harish et al., 2014), Task allocation (Kang et al., 2013), Parallel machine scheduling (Ying et al., 2010), Freight train scheduling (Yuan et al., 2008), Non-permutation flow shop scheduling (Ying, 2008), and Flow shop scheduling (Ruiz et al., 2008). To the best of our knowledge, the IG algorithm has not been used in the RSP before. This is the first study to use local search and biased randomisation to improve the IG algorithm for the RSP with deterministic delays.

- Variable Neighbourhood Search (VNS) and Biased Randomised Variable Neighbourhood Search (BR-VNS)

The VNS has been used very successfully to solve some types of optimisation problems such as Capacitated location routing (Derbel et al., 2011), Job shop scheduling (Liao et al., 2007; Roshanaei et al., 2009; Zandieh et al., 2010), and Vehicle routing (Bräysy, 1999; Polacek et al., 2005). To the best of our knowledge, this is the first study to use a biased randomised technique to improve the efficiency of VNS for solving the RSP with regard to delays.

C. Optimisation model for the stochastic UK-RSP and Thai-RSP

An optimisation model was proposed for solving the UK-RSP and the Thai-RSP using stochastic delays that randomly occurred in the network. The objective is to minimise the total delay of all trains considered by two types of signalling constraint.

D. Sim-optimisation solution methods for the stochastic UK-RSP and Thai-RSP

- Sim-Iterated Greedy with Biased Randomised (S-IG-BR) and Sim-Biased Randomised Iterated Greedy with local search (S-BR-IG-LS)

The combination of Monte Carlo Simulation (MCS) and IG with biased randomised or Biased Randomised IG with local search are also called Sim-IG with biased randomised or Sim-Biased Randomised IG with local search. To the best of our knowledge, these two methods have not yet been used to solve the stochastic RSP in the literature; this is the first study which uses S-IG-BR and S-BR-IG-LS to solve the stochastic RSP. In addition, the MSC was used to prove the reliability of IG-BR and BR-IG-LS.

- Sim-Variable Neighbourhood Search (S-VNS) and Sim-Biased Randomised Variable Neighbourhood Search (S-BR-VNS)

Some researchers used the Variable Neighbourhood Search (VNS) to solve the stochastic RSP to deal with unexpected delays. However, in the best of our knowledge, no one has adapted Monte Carlo Simulation (MCS) with the VNS algorithm before, so this is the first study used the combination between MCS and VNS or Biased Randomised VNS which can be called Sim-VNS and Sim-Biased Randomised VNS to solve the stochastic RSP. Then, MSC was used to measure the reliability of VNS or BR-VNS.

1.4. Thesis Structure

This thesis is divided into seven parts (excluding the references) covering discussion on the topic of RSP regarding both deterministic and stochastic delays. Basic knowledge was provided related to the RSP under disturbances or disruptions with focus on using the optimisation model to implement the new algorithm for solving the RSP. Chapter contents are described below:

Chapter l - This chapter provides the introduction, motivation, contribution, aims, objectives and structure as the overview of the thesis.

Chapter 2 - Literature reviews related to the RSP are presented concerning the railway planning process, types of RSP, solution methods, topology, explanation of disruptions, signalling systems and a tree search graph method comprising the main focus points.

Chapters 3 to 8 - This thesis can be separated into two main parts each having three chapters (six chapters in total). In each part, an explanation of the RSP optimisation model, basic knowledge of the algorithm and implementation based on different algorithms are used to solve real-world data from UK and Thai railways. The experimental results are

Part I: Deterministic RSP in the presence of uncertainties

Chapter 3 - Iterated Greedy with Biased Randomised Algorithm for Deterministic RSP

Chapter 4 - Biased Randomised Iterated Greedy with Local Search Algorithm for Deterministic RSP

Chapter 5 - Biased Randomised Variables Neighbourhood Search Algorithm for Deterministic RSP

Part II: Stochastic RSP in the presence of uncertainties

Chapter 6 - Sim-Iterated Greedy with Biased Randomised Algorithm for Stochastic RSP

Chapter 7 - Sim-Biased Randomised Iterated Greedy with Local Search Algorithm for Stochastic RSP

Chapter 8 - Sim-Biased Randomised Variables Neighbourhood Search Algorithm for Stochastic RSP

Chapter 9 - This chapter presents the conclusion and recommendations for future studies.

1.5. Chapter Conclusion

This chapter provided the background, motivation, aims, objectives and contribution and showed the structure of the whole thesis. Furthermore, it also developed a clear understanding of the thesis structure. The literature review investigated the deterministic and stochastic RSP, while characteristics of RSP were used to develop an optimisation model. Then, the IG and VNS algorithms were adapted to solve the RSP. Finally, the experimental results were discussed to assess the efficiency of the algorithms.

Chapter 2: Literature Review

2.1. Introduction

This chapter focuses on the available literature regarding the train scheduling problem and solution methods used to solve the problem and improve solution quality. Firstly, this thesis provides some knowledge on the planning process to increase understanding about how it is undertaken. Secondly, network topology and signalling systems are provided to describe the problem. Then, the previous studies on deterministic and stochastic delays relating to the railway scheduling problem are presented. Moreover, the classification of disruptions, impacts of disruptions to the rail system and ways to handle the disruptions are presented to show strategies which can be used to improve and regenerate a new timetable. Finally, the solution methods usually used to solve the RSP are provided.

2.2. Railway planning process

Academics have solved railway transportation problems by using operation research techniques since the 1950s (Beckmann et al., 1955). Jespersen-Groth et al. (2009) investigated disruption in railroad transportation and separated their research into three subtopics as train scheduling, crew scheduling and rolling stock management. The main train transportation problem focused on here is the train scheduling problem. All trains in a rail network share the same infrastructure (Khosravi, 2013), therefore the company requires an efficient timetable to manage the time slot for every train which also shows the normal operation time for passengers to track and plan their journey. However, many incidents can affect the timetable. If they occur, the original train timetable will break or require change (Nielsen et al., 2012).

Normally, railway companies design their process by using a hierarchical structure to divide the required decision-making (Lusby et al., 2011; Bussieck, 1998). According to Figure 2.1, the planning process required before the company can create a train schedule is shown as three different levels which are: strategic level, tactical level and operational level (Assad, 1980; Huisman et al., 2005; Lusby et al., 2011). Each level has a different purpose to generate train scheduling. If something goes wrong in the process, the planner has to return to the previous stage and re-plan it (Lusby et al., 2011; Bussieck et al., 1997). Details of each level are as follows:

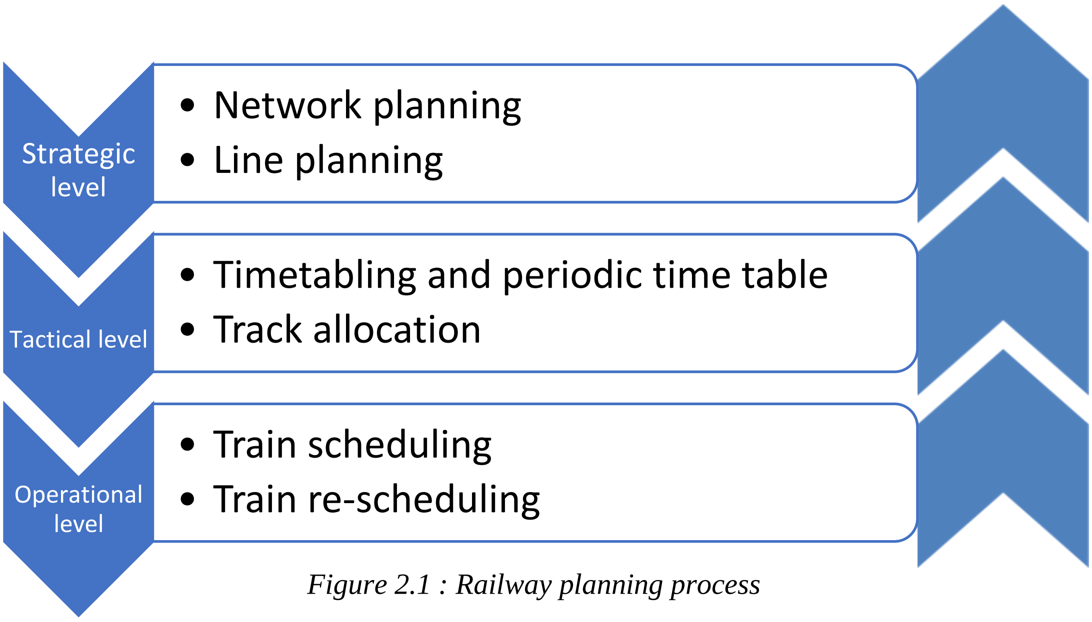

Figure 2.1 : Railway planning process

2.2.1. Strategic level

The strategic level is a long-term plan which focuses on acquisition and infrastructure and will be used for around 5 to 15 years (Borndörfer et al., 2012; Bussieck, 1998). This level of the process is used to check up and reserve the resources such as infrastructure, rolling stock and crew because all of these resources are shared among the rail system (Ralf Borndörfer et al., 2012). Moreover, the planning used at this level is mainly network planning and line planning (Lusby et al., 2011; Borndörfer et al., 2012).

2.2.1.1. Network planning

Network planning considers the basic railway infrastructure and company plans for construction and/or modification of the existing setup (Schlechte, 2011a; Lusby et al., 2011). Railway infrastructure consists of two main parts as the station or junction and the line. The station or junction is where all trains and passengers interact, while the line connects two junctions (Lusby et al., 2011). Moreover, the network planning phase is used to check the availability of the infrastructure. Minimisation of construction cost is the goal of network planning (Kinder, 2008).

2.2.1.2. Line planning

Line planning selects a route for each train, for example, train origin, train destination, stops, links and cycle time (Bussieck, 1998). Train origin and destination refer to the start point and end point of each line; stops are stations where the train receives passengers between the origin and destination station. Links are crossovers to other lines and cycle time is the frequency of trains that should be operating within the specific time period (Lusby et al., 2011). Furthermore, the decision to select a line for a train should be based on passenger demand which is calculated using statistics to forecast the number of passengers between the train origin and the train destination (Bussieck et al. , 1997; Schlechte, 2011a). Therefore, the main objective function for the line planning problem is to maximise passenger satisfaction (Lusby et al., 2011).

2.2.2. Tactical level

The tactical level is a mid-term plan which concerns allocation and management of all resources for each train; it can be used for 1 to 5 years (Borndörfer et al., 2012; Schlechte, 2011b) . In this level, each station manages a fixed arrival time, departure time and the tracks for each train (Lusby et al., 2011). This plan can be used to generate a complete time schedule. The tactical level focuses on timetabling and track allocation problems which are necessary to allocate times and tracks for creation of schedules in the next stage (Tornquist, 2006).

2.2.2.1. Timetabling and periodic timetable

Caprara et al. (2007) stated that after a company sets the route and cycle time of each train in the line planning problem, the time slot of each train can be generated, called the train timetabling problem. The train timetabling problem is concerned with setting an arrival time and departure time for each train and reserving it to generate the full schedule (Schlechte, 2011a; Borndörfer et al. , 2012) . In addition to creating the timetable, the planner needs to allow for safety requirements such as the headway time between trains using the same track and crossing of train in opposite directions (Cacchiani et al., 2012; Kinder, 2008; Schlechte, 2011a) . This problem can also be categorised as a cyclic and noncyclic timetable. A cyclic timetable has a fixed cycle time period, while the noncyclic has more flexible time slots depending on time periods (Khosravi, 2013). Consequently, the train timetabling problem is used to provide the time slot of each train which meets the safety rules for the train company and make decisions to build train scheduling.

Caprara et al. (2007) claimed that most cyclic timetabling is based on the periodic event scheduling problem (PESP) introduced by Serafini et al. (1989). The PESP is an event which occurs only in a small period of time in the cyclic timetable (only 1 or 2 cycles) and then the normal cycle is repeated (Cacchiani et al., 2012). This event is the set of arrival time, departure time and cycle time as different from the normal timetable to satisfy increasing passenger demand (Kinder, 2008; Huisman et al., 2005). For example, during the rush hour or peak time of the day, the PESP will operate more trains than at other times (Cacchiani et al., 2012; Kinder, 2008). Furthermore, PESP has an objective function to minimise passenger dwell time at the station (Cacchiani et al., 2012; Huisman et al., 2005). Therefore, the PESP is used to improve satisfaction of the passengers.

2.2.2.2. Track allocation

After planning a route and creating a timetable for each train, the company deals with track allocation by using the same constraints about safety factors. The track allocation process is very similar to the timetabling process. The former is used to allocate a slot of time and the latter is used to allocate a slot of track (Schlechte, 2011a). Moreover, some logisticians claim that the track allocation problem and train timetabling problem are similar; they combine these two problems as the train timetabling problem (Borndörfer et al., 2007; Brännlund et al., 1998; Caprara et al., 2001; Caprara et al., 2002). However, some authors separate these facets into two sub-problems (Kroon et al., 2008; Schlechte, 2011a; Lusby et al., 2011). Furthermore, this step focuses on reserving an available track for trains by using the input of line planning and timetabling from the previous process (Borndörfer et al., 2012). For track allocation, trains should not overlap with each other if they operate following the timetable since one section of track cannot be used by more than one train at the same time. Track allocation and timetabling can be more flexible for cargo or freight rail operations because the arrival time and departure time are only important for the station loading or unloading the items. Parkes and Ungar (2001) classified type of track as single-track, double-track and yard section. These differences also relate to the complexity of a problem and may require advanced constraints for special characteristics. A single-track is widely used for most journeys as the basic which considers only one track each time. A double-track refers to a pair of tracks used for operating a train in the same direction and allowing trains to pass each other. A yard section refers to a pair of tracks which are used for operating a train in the opposite direction and allowing trains to meet each other. In general, rail networks use double-track and yard section more than single-track. Therefore, track allocation is the last process to create time scheduling at the operational level.

2.2.3. Operational level

The operational level is a short-term plan which focuses on creating train scheduling, monitoring and then waiting for a response in real-time situations; this plan can be used for a day or up to a year (Borndörfer et al., 2012; Schlechte, 2011b). In this stage, a train schedule is generated following the first two levels of planning to provide some response to solve real-time disruption management (Khosravi et al., 2012). At this level, problems mainly focus on train scheduling and train re-scheduling.

2.2.3.1. Train scheduling

The original train schedule is generated using the output from the first two levels as network planning, line planning, timetabling and track allocation (Khosravi, 2013). The train schedule provides a train routing which shows information about the stations between the start and destination point and presents a timetable which shows the arrival time and departure time at each Station. This stage can be called the planning decision stage because it combines all the outputs to create a schedule which the train company can use for a month or a year (Huisman et al., 2005; Schlechte, 2011a). The main purpose of the train schedule is to manage the rail network and show journey details to the passengers. Firstly, the train schedule is used to control all trains in the network for optimal operation without interfering with other trains (Corman et al., 2010; Gulati et al., 2016). Secondly, it provides details of each journey for the traveller to use to plan ahead or track the location and status of each train (Tornquist, 2006). Moreover, the objective function of the train scheduling problem is to minimise the total running time of all trains in the system by considering the characteristics of the rail network. Therefore, the train scheduling process is very important for train management to improve efficiency of the railway operations system.

2.2.3.2. Train re-scheduling

Train rescheduling is the last process which focuses on adjustments to the train schedule which are only made on a daily basis to solve real-time disturbances or disruptions such as weather condition, accident or crew no show (Khosravi, 2013). Management monitors how the trains operate following the original schedule in a real-time situation. If the rail network is disturbed by an unexpected event, then the rescheduling process is necessary to generate a new schedule (Caimi et al., 2012; Törnquist et al., 2007; Hofman, 2005).

However, the train schedule is reset back to the original plan for the next day. The rescheduling process is worked out by going back to the previous levels (strategic level and tactical level) and using the original plan as input to try to avoid delay propagation (Cacchiani et al., 2012). The main purpose of the train rescheduling problem is to minimise total train delay in a limited time with a small adjustment (Huo et al., 2016). Train rescheduling has an objective function to minimise total delays when disturbances or disruptions occur; however, the solution is not optimal because of the time limit. The new scheduling should be regenerated as quickly as possible to ensure that train operation is continuous without freezing the rail system (Lusby et al., 2011). Moreover, if the new schedule is similar to the original schedule, then this can improve customer satisfaction and reliability (Huo et al., 2016). Therefore, the rescheduling process is used to create a new timetable to manage unpredictable events that break the original time schedule.

2.3. Network Topology

Normally, RSP complexity can be classified using two main points as (i) size of network and infrastructure (topology) and (ii) trains traffic or disruptions. The network topology can be classified into three levels of detail concerning rail infrastructures as microscopic, mesoscopic or macroscopic problems (Caimi, 2009).

- Microscopic problems provide more local detail about track and train operations. This level of RSP planning can generate more reliable solutions.
- Mesoscopic problems miss some details of track and operation at the local level. Missing data is assumed by using data from other standard networks. Therefore, the results only show an overview of the planning process.
- Macroscopic problems only consider global interdependencies over the whole rail system using a simplified safety model without detail on a local level. Normally, macroscopic timetables can be used as input for the microscopic level.

Here, we selected network topology depending on the characteristics and propose of the RSP because each type has advantages over the other. In this thesis, we focused on microscopic level scheduling which considered track and train details. Output scheduling provided details of time and order of the trains on each track section.

2.4. Signalling system

Pachl (2009) studied signalling systems and rules of railway operations. He classified a signalling system into two principles dependent on number of blocks of information, named as one-block signalling and multiple-block signalling. In addition, track sections of railway are divided by signal and the section between two signals is called block. In general, each country has their own signalling regulation dependent on the railway infrastructure characteristics. The signalling system is one of the most important components of a railway transportation system to improve the safety.

2.4.1. Two-aspect signalling

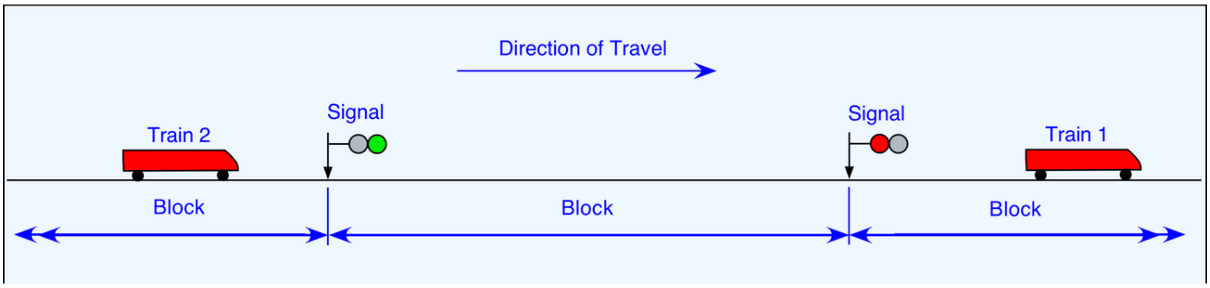

Figure 2.2 : Two-aspect signalling (The Railway Technical Website, 2018)

Two-aspect signalling only provide information two blocks ahead. This type of signal includes only two colours which are red for stop (danger) and green for clear. When a train has entered a block, the signal turns red. This means that other trains should stop and wait until the previous train leaves that block, then the signal will return to green. This type of signalling is commonly used in Thailand (Figure 2.2).

2.4.2. Four-aspect signalling

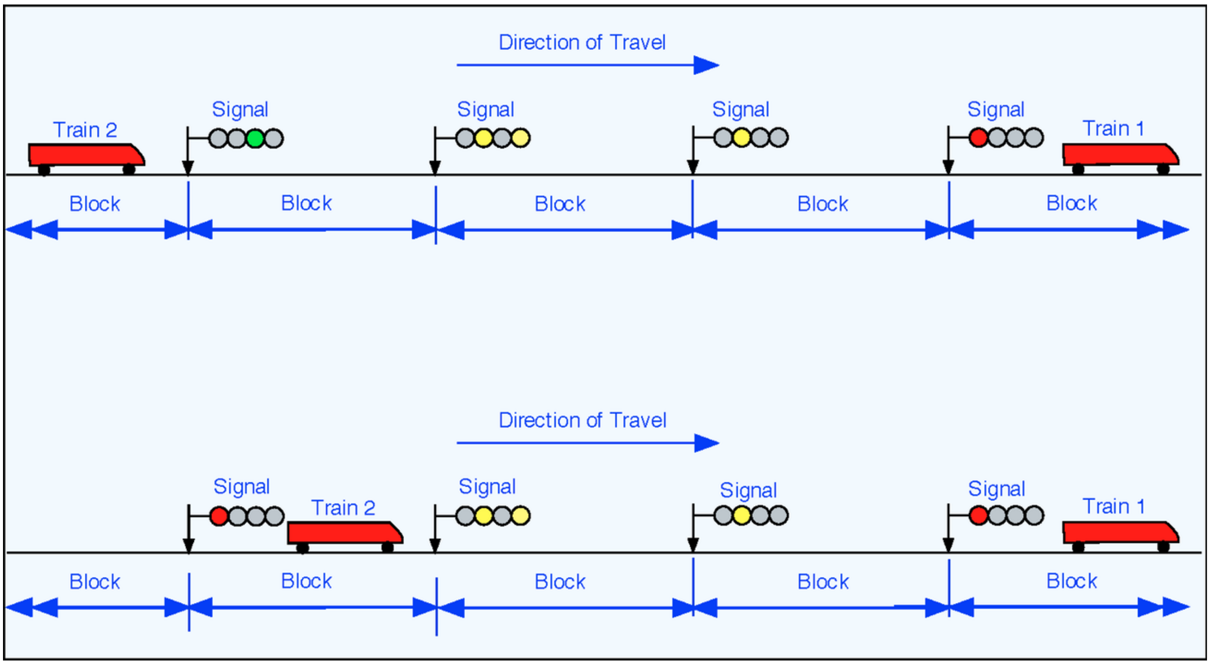

Figure 2.3 : Four-aspect signalling (The Railway Technical Website, 2018)

Four-aspect signalling is commonly used in UK railway networks. In Figure 2.3. the status of the next four blocks are shown as a signal colour, this means that train drivers can adjust the speed of the train and safely brake if an accident occurs with the previous train because the signalling system provides distance gaps between two trains which match safety regulations. In the four-aspect signal, the signalling information of each colour is provided as follows (i) red for stop (danger), (ii) yellow for caution, (iii) double yellow for preliminary caution and (iv) green for clear. Therefore, all trains in the UK system network travel at a fixed speed to keep more than three blocks away from each other.

2.5. The railway scheduling problem (RSP) under the uncertainties

Many researchers studied the railway scheduling problem with uncertainties and separated it into two sub-topics depending on their input data as the deterministic railway scheduling problem (Deterministic RSP) and the stochastic railway scheduling problem (Stochastic RSP). These uncertainties include delays when the train travels, dwell time or late departure from the station.

2.5.1. Deterministic RSP

In the deterministic, input data is known before the start to find a result. Thus, running time, dwell time and late departure time delay should be known in advance using historical data and known information about the infrastructures. The deterministic scheduling problem has been widely studied since the mid-1950s for different types of problem, method and data sizes (Potts et al., 2009).

Szpigel (1973) was the first to use a linear integer program and branch and bound algorithm for the deterministic railway scheduling problem. The objective function of this paper was to minimise total running times by using only a small test case.

Kraft (1987) developed a deterministic train dispatching approach to minimise total delay based on train priority. This research aimed to regenerate a dispatching rule for the train when the delay occurred.

Şahin (1999) focused on train scheduling based on inter-train conflict management with the aim to minimise the overall delay in the train system. A consequential delay was considered that propagated from other trains. The consequential delay was calculated from delay formulation and termed as the first train delay value. A heuristic algorithm was also developed to find a good solution in the shortest computation time.

Liu et al. (2009) developed train scheduling by using a blocking parallel-machine job shop for both single track and double tracks. They used a case study from Queensland Rail to set the test instances with additional running time caused by unexpected events. The author provided a Shifting Bottleneck Procedure (SBP) and Feasibility Satisfaction Procedure (FSP) solution method to minimise the total delay.

Gafarov et al. (2015) studied a single-track railway scheduling problem whereby the test case only considered two stations. Two sets of trains run in opposite directions and all trains have equal speed. They stated that a single-track railway is a bottleneck point because two trains cannot run in opposite directions on a single track; when the first train is travelling on the track the second train must wait. This model was developed to solve and minimise the total delay.

Samà et al. (2017) considered how to minimise train delay and avoid deadlock using 20 instances for each test case with different initial delays from Italian, Dutch and British railways. The problem model used a mixed integer linear program (MILP) and metaheuristics to improve solution quality.

Corman et al. (2017) proposed MILP to solve the railway traffic scheduling model and reduce delay in railway networks. They focused on rescheduling of real-time traffic management. The experimental results were based on two case studies with a set of deterministic delays and implemented using CPLEX and four heuristic based algorithms.

Khadilkar (2018) used a reinforcement learning (RL) approach for bidirectional railway scheduling using both single-track and multi-track. The goal of this model was to minimise total weight tardiness and then compared results from the RL algorithm to other heuristics using data from the Indian rail network to reach the real-world purpose.

2.5.2. Stochastic RSP

The stochastic RSP has only been considered in the last ten years because it is more complex and harder to generate stochastic data. Random delays were generated using probability distribution, simulation and historical trends. The model used these values to solve and provide more realistic results. The main advantage of the stochastic RSP is to simulate the real environment whereby uncertainties randomly come into the system and then check that the model and algorithm work properly when used in a real situation.

Rodriguez (2007) proposed a constraint programming model for the train routing and scheduling problem and used this model to find an alternative route to reduce the overall delays. The simulation program randomly added delays to the input data sets as test instances. The program ran for only 180 seconds of computation time for rescheduling.

Kroon et al. (2008) proposed a stochastic optimisation model to improve the efficiency of railway scheduling with stochastic disruption. The stochastic model works by adding time supplements and buffer times into the timetable. These can make a schedule more robust and absorb some stochastic disruption.

Khan et al. (2010) developed a two-stage stochastic optimisation model to solve a real-time dispatching problem. They proposed minimising total running time and overall delays by considering the stochastics delay based on the Beijing-Shanghai high-speed rail in China.

Amraoui et al. (2014) proposed a scenario-based stochastic and mixed linear programming model to solve the railway scheduling problem under the uncertainty of train running time. In their paper, a delay was randomly added to the transportation time of each train based on historical data. The objective function was to minimise the makespan.

Pellegrini et al. (2014) considered optimal train routing and scheduling with MILP. The model proposed to minimise the delay when unpredictable events occurred. Moreover, the authors used two case studies from a complex junction in France to determine both deterministic and stochastic scheduling. For the stochastic situation, they randomly generated the delay based on a triangle in the Gagny area and used real data from the Lille-Flanders Station for a deterministic approach.

Abid et al. (2015) studied a railway scheduling problem of a single line track using the job shop scheduling model and branch and bound algorithm. Their model proposed minimising total running time using an exact lower bound rule to estimate stochastic train delays. Moreover, data from a real-world case study were collected from Rawalpindi to Lalamus, Pakistan.

Rudan et al. (2019) applied a MILP model to reduce delay using dynamic traffic management to reschedule trains. Experiments were performed on the Dutch railway system, Netherland which consisting of 41 stations and 118 tracks. Network delays followed a Weibull distribution with 2 cases where 8% and 20% of the trains were selected to be delayed. Results were compared with other solver programs including DIP, SYMPHONY, CBC, GLPK and CPLEX. The CBC solution outperformed all the others.

2.6. Classification of Disturbances/Disruptions

Normally, the train will operate following the timetable that is created by a specialist. Here, this is called the original timetable. This timetable will not change unless an unexpected event occurs; we called that event the disturbance and disruption. Disturbance and disruption usually occur on a rail network and all delays cannot be avoided.

Disturbance and disruption can be classified into two types as the causes of the problem which are primary delays and secondary delays.

Primary delays occur directly from disturbance or disruption (Shafia et al., 2012). This type of delay is caused by an unexpected event which occurs during the rail operation which cannot be avoided (Keita et al., 2016). However, a reliable schedule can be provided by understanding the primary delay. The way to do this is to analyse the characteristic of each disturbance or disruption and leave gaps in the timetable that can absorb small-sized delays. If the delays are too large this will require repairing/recovering or rescheduling (Hofman & Madsen, 2005). Examples of primary delays are accidents, weather conditions, disasters, failures of the infrastructure and unavailability of resources.

After primary delays have occurred, they may spread to other trains in the same network such as a later train on the same track (Shafia et al., 2012) and the connection line due to interaction activities in the rail network (Keita et al., 2016). This can be called a secondary delay or knock-on delay (Jespersen-Groth et al., 2009). This delay can be avoided or solved by adjusting the scheduling in real-time. The event has already started and we already know the delay (Keita et al., 2016). Moreover, secondary delays can also be prevented by generating a reliable schedule; this concept is provided by the slack time in the timetable to limit the propagation of delay to another train (Hofman & Madsen, 2005; Keita et al., 2016; Shafia et al., 2012).

2.6.1. Accidents/Collisions

The most common disturbance types which occur in the rail process and can found daily are accidents and collisions (Huo et al., 2016; Andersson et al., 2013). These two basic disturbances provide a short-delay and small effect to the original timetable; however, both have a different problem source.

Accidents are usually caused by activities of people that work in the rail network operation. Normally, activities that deal with people cannot be completely controlled because every person has a different reaction when doing the same thing. This can be considered as human error (Yin et al., 2016; Hofman, 2005; Jespersen-Groth et al., 2009). Moreover, this problem can be separated into two sides as passenger error (Andersson et al., 2013) and rail network staff error. Both of these are very similar and delay the train for only a few seconds or a few minutes (Yin et al., 2016). This type of disturbance is caused by itself or considered as a primary delay, easy to solve and can prevent the reliability of the timetable during the planning process (Andersson, 2014).

Moreover, collisions are not caused by the process of an individual train but through interaction with other trains in the rail network which share the same infrastructure (Andersson et al., 2013). These can be called secondary delays or knock-on delays (Jespersen-Groth et al., 2009). This disturbance type is not hard to solve but it is hard to know when it occurred until the event starts because when one train in the system is delayed, it will affect other trains in the same line and their connection lines (Salido et al., 2008; Hofman, 2005). The infrastructure is limited and cannot be used at the same time (Yin et al., 2016).

2.6.2. Weather conditions

The majority of people accept that weather conditions cause large disturbances or disruptions for rail networks. Weather problems cannot be prevented and can cause short-delays, long-delays or cancellation of the train (Hofman, 2005; ; Khosravi, 2013; Amraoui et al., 2014; Corman et al., 2011; Tamannaei et al., 2016; Jespersen-Groth et al., 2009; Meng et al., 2011; Veelenturf et al., 2009; Huo et al., 2016; Xia et al., 2013). Weather conditions can be separated into two parts using the weight of the problem as regular weather condition and extreme weather condition (Rossetti, 2007). Weather condition normally affects the speed of the train; sometimes the train driver has to reduce speed to meet security requirements (Amraoui et al., 2014). However, some weather conditions also stop rail operation such as blocking rail tracks or vision. If this occurs we need to wait until it is cleared (Rossetti, 2007).

2.6.2.1. Regular weather condition

Regular weather conditions have minimal effect on the train process and cause only minor delays. Examples of normal weather conditions are rain, light snow, low temperature (Budai-Balke, 2009), high temperature (Budai-Balke, 2009; Dobney et al., 2009; Rossetti, 2007) and strong wind (Xia et al., 2013; Rossetti, 2002). This type of disturbance may require decreasing the speed of the train to be safe for passengers and not exceed the maximum speed level. Sometimes this may increase train travel duration from the estimated time in the schedule (Amraoui et al., 2014; Rossetti, 2007). Normally, security requirement rules are used to control train speed for safety reasons and protect passengers from unexpected accidents. The maximum speed that meets with security rules is dependent on path and train model which have different specifications (Hofman, 2005; Harrod, 2006; Yang et al., 2016). A train reduces speed because of problems such as slip, heat and air resistance (Rossetti, 2007). Track slip can be caused by rain (wet), snow (wet and/or ice) and low temperature (ice) and result in the train wheels working in a non-effective way (Xia et al., 2013; Amraoui et al., 2014). High temperature also affects the railway because heat from friction between wheels and track can cause fire and destroy the infrastructure (track and wheels) (Dobney et al., 2009). Strong winds can be split into two sub-types as air resistance and air support depending on train direction. All types of strong wind may affect the stability of the train and the driver should reduce speed (Xia et al., 2013). All regular weather conditions only cause train delays for a few minutes or a few seconds. These delays can be prevented by providing an extra gap of time in the train timetable (Schlechte et al., 2009; Cacchiani et al., 2012).

2.6.2.2. Extreme weather condition

Extreme weather conditions can result in large problems or major delays to the timetable and cause long train delays and cancellations. Rainstorms and snowstorms are the main extreme weather conditions which occur in real-world situations. When this type of disruption occurs, we can only wait until it stops or decreases in level and everything returns to normal. The reason why this disruption has a very large effect on the rail network is because it completely blocks the driver's vision and track, which results in a train shutting down their journey immediately (Rossetti, 2007). The driver's vision is very important when the train is running on the track, because a driver needs to know what is going on in front of the train and whether the way is clear or not. When a storm occurs, the driver cannot see forward more than three or four metres; if a tree, vehicle, rock, animal or people are on the track, then the driver cannot react quickly. This may cause a big accident and be very dangerous for passengers (Corman et al., 2011). When the track becomes blocked by debris from storms or snowdrifts, the rail network must stop operations until the way is clear. They must use an alternative route or provide other transport arrangements for passengers to complete their journey (Rossetti, 2007). We cannot prevent this disruption and can only prepare solutions to solve these problems as smoothly as possible.

2.6.3. Disasters

Natural disasters directly damage overall rail network infrastructure including track and station (Rossetti, 2002). This event occurs suddenly and provides large disruptions to the train scheduling, resulting in train cancellation or long delays which require rescheduling. Examples of natural disasters are earthquake, flood, mudslide, stonefall, fire and volcanic eruption (Yin et al., 2016; Southwell, 2012). These issues have no specific solution but some can be predicted or forecast and a warning message can be provided to the passengers.

Levels of disruption which affect the rail infrastructure start from blocked tracks, broken tracks or completely destroyed tracks (Yin et al., 2016). Blocked tracks can be quickly repaired by railway staff but broken or destroyed main infrastructures require rail engineering to fix or re-build (Southwell, 2012). The company loses revenue, but reliability will not be lost through these unfortunate events if there are proper processes in place to solve the problem. Natural disasters cannot be predicted but we can learn how best to deal with them (Azad et al., 2016; Southwell, 2012).

2.6.4. Failures due to the infrastructure

The infrastructure is the most important part of the rail operation system. Trains cannot run without infrastructure. If this is limited, there may be no alternative route for continuous running of the train (Samà et al., 2017). Examples of infrastructure failure are signal failure, a break in the overhead line, error of train engine or broken tracks (Veelenturf et al., 2012). This type of disruption results in a long-time delay because when it occurs there are only two solutions to the problem. The first is using an alternative route to drop passengers at their destinations and the second is to wait until engineering can fix the problem (Nielsen et al., 2012). Both ways usually take a long time to solve. Trains can be delayed more than an hour or even cancelled (Yin et al., 2016). Therefore, these disruptions require large changes in the timetable by rescheduling all trains on the same route. Furthermore, the duration of the problem is not known and can only be estimated using historical data. Rescheduling requires real-time information (Jespersen-Groth et al., 2009). Moreover, this type of disruption can be minimised by regular maintenance of all infrastructure. However, it is difficult to plan maintenance work at high cost (Budai-Balke, 2009).

2.6.5. Unavailability of resources

Two main resources that are important for the railway process are crew and rolling stock. Lack of either of these resources can be the cause of train delays or cancellations and generate disruption (Veelenturf et al., 2009; Nielsen et al., 2012). Unavailability of resources which can cause disruptions include management of crew member scheduling such as crew sickness (Jespersen-Groth et al., 2009) or delays of crew and crew strike action (Veelenturf et al., 2012). Crew members are vital for smooth operation and lack of staff causes delays (Jespersen-Groth et al., 2009). However, large disruptions are caused by infrastructure failure such as lack of rolling stock locomotives, carriages, wagons and wheels tracks (Azad et al., 2016; Khosravi, 2013). Lack of rolling stock can increase the duration of engineering work Crew scheduling and rolling stock management are vitally important (Nielsen et al., 2012).

2.7. Impact of Disturbances/Disruptions to the rail system

Normally, trains operate following a timetable that is created by a specialist; we call this the original timetable. This timetable will not change unless an unexpected event occurs; this event can be called a disturbance or disruption and is classified by impact types as size of delay, either minor or major (Veelenturf et al., 2009). Both types of delay affect the train schedule and result in the train not arriving on time. There are many ways to solve this problem such as repairing/recovering ((Huo et al., 2016; Hofman, 2005; Liebchen et al., 2009), and rescheduling (Tamannaei et al., 2016).

2.7.1. Minor delays

A minor delay is a small-scale delay which occurs during train operation. It can be called a disturbance (Azad et al., 2016). Minor delays are usually caused by people such as passenger error or driver error; however, weather is also one of the causes of this delay type. When the disturbance has occurred, the train schedule must be adjusted. Only a small change is enough because the delay is just a few seconds or a few minutes (Yin et al., 2016). In this type of delay, some of the effects of disturbances can be prevented in the original timetable by using forecasting or probabilistic based on historical data, providing some extra gaps of time in the timetable (Salido et al., 2008; Usida et al., 2011) or repairing/recovering (Huo et al., 2016; Hofman, 2005; Liebchen et al., 2009).

2.7.2. Major delays and cancellations

A major delay is a large-scale delay that causes a stop in the train process (Nielsen et al., 2012). A major delay can occur suddenly and cannot be forecasted. Examples of this type of delay are strong weather conditions, disasters, infrastructure failures and unavailability of resources (Veelenturf et al., 2012). When the disruption has occurred, the rail network suffers a long-time delay and this also affects other routes in the same network (Jespersen-Groth et al., 2009). This type of delay requires a rescheduling process (Tamannaei et al., 2016). Regular maintenance with good crew and stock management can mitigate some disruptions caused by infrastructure failures and unavailability of resources (Veelenturf et al., 2012).

2.8. How to handle the Disturbances/Disruptions

Many logistics specialists offer scheduling strategies to solve the railway stochastic scheduling problem and aim to generate a train timetable that can deal with the uncertainty. These strategies show how a planner can handle the disturbance or disruption event and they can be classified into four categories (Van De Vonder et al., 2007; Ouelhadj et al., 2009; Amraoui et al., 2014; Carvalho et al., 2016; Lambrechts et al., 2008; Fischetti et al., 2009; Bertsimas et al., 2010; Wilson, 2016; Salido et al., 2008; Li et al., 2016; Shafia et al., 2012) as Reactive scheduling, Proactive scheduling, Preventive maintenance and Robust optimisation. These are described in more detail below.

2.8.1. Reactive scheduling

Reactive scheduling is used when a timetable is broken by disturbance or disruption activities. This strategy is used to generate a new timetable after collating all the information about what actually occurred to minimise train delay (Amraoui et al., 2014). Furthermore, when an unexpected event occurs, decisions for rescheduling or repairing the schedule are made immediately (Carvalho et al., 2016; Lambrechts et al., 2008). Planners need to make decisions to generate new schedules as fast as possible. The priority dispatching rule is frequently used to make a new schedule by first selecting a train which has high priority. This rule is very popular because it is easy to use and gives a reasonable solution in short execution time (Ouelhadj et al., 2009). This category has characteristics that focus on real-time reaction with the uncertainty (Van de Vonder, 2006) and unexpected events are not considered before they occur (Carvalho et al., 2016).

2.8.2. Proactive scheduling

Proactive scheduling focuses on building the train schedule using historical information to predict uncertainty events through statistical knowledge, also known as “Predictive scheduling” (Van De Vonder et al., 2007; Lambrechts et al., 2008). Proactive scheduling aims to minimise the makespan or traffic frequency (Amraoui et al., 2014) and increase the stability of the timetable (Lambrechts et al., 2008). It should be able to absorb future disturbance or disruption and also make the timetable more reliable (Van de Vonder, 2006)). Unfortunately, proactive scheduling cannot protect against all delays that occur which need to be managed when the timetable is broken (Lambrechts et al., 2008). Many logisticians suggest combining reactive and proactive scheduling as "Proactive- reactive scheduling" to solve the RSP (Van de Vonder, 2006; Lambrechts et al., 2008). Proactive-reactive scheduling creates the original schedule through proactive planning approaches but uses the reactive policy to undertake rescheduling or repairing (Van De Vonder et al., 2007). Therefore, proactive scheduling is often combined as proactive-reactive to improve the quality of train scheduling and rescheduling for real-time unpredictable events (Van de Vonder, 2006; Ouelhadj et al., 2009). Ouelhadj et al. (2009) suggested that proactive-reactive scheduling might sometimes provide a new timetable which is significantly different from the original and decreases reliability for passengers.

2.8.3. Preventive maintenance

Preventive maintenance is not directly used to create rail scheduling but to increase the efficiency of the rail network and decrease the probability of failures due to the infrastructure as the cause of some major delays (Jespersen-Groth et al., 2009; Budai-Balke, 2009). Most major delays are caused by lack of infrastructure or unavailability of resources. These failures can be prevented by pre-planning maintenance schedules to fix repair infrastructure before it fails (Jespersen-Groth et al., 2009) and manage the rolling stock (Van de Vonder, 2006). There are many ways to resolve hidden problems by repair or replacement to discover which spare parts are required. Pre-planned maintenance can improve the reliability of the train company and inform passengers in advance of any expected delays or schedule changes (Budai-Balke, 2009). Preventive maintenance is considered as one strategy to deal with disturbance or disruption by preventing the cause of the problem before it occurs.

2.8.4. Robust optimisation

In the last few years, many logisticians have considered ways to make train timetables more reliable and prevent small delays. Robust optimisation was introduced to make a strong timetable that can absorb unexpected events which occur during the train operation process (Salido et al., 2008). Robustness can be defined as a train timetable that can deal with an unexpected situation without significant adaptation to the original plan (Takeuchi et al., 2005). However, this case only works with a small disturbance or a low level of delays. Normally, the main type of delay which occurs daily are minor delays and these can be absorbed by a strong, robust timetable; however, medium to large scale disruptions still require a rescheduling process to generate a new timetable (Li et al., 2016; Fischetti et al., 2009). The majority of research studies measure the degree of robustness stability using many factors or methods to compare between the original timetable and the simulated timetable and then focus on minimisation of tardiness to produce the solution (Fischetti et al., 2009; Bertsimas et al., 2010; Wilson, 2016; Salido et al., 2008). Only a few researchers used this concept to generate a robust timetable and improve the quality of service of the train company (Li et al., 2016; Shafia et al., 2012).

In addition, if the company needs to get a stable time schedule, they should be considering a trade-off between the solution performance and timetable reliability (Bertsimas et al., 2010). Salido et al. (2008) suggest three main methods that can be applied to generate a robustly optimised timetable are (i) decreasing optimisation, (ii) decreasing capacity, and (iii) decreasing heterogeneity.

2.8.4.1. Decreasing optimisation

Some small disturbances do not allow the travel duration to be calculated using maximum speed or slot time because when a minor delay occurs it will directly affect the network (Salido et al., 2008). The robustness of the timetable will be increased by providing some gap or extra time slot as a buffer time (Andersson et al., 2013; Azad et al., 2016). The buffer time will help the timetable to absorb small scale delays, as a few minutes or a few seconds and allow the train to follow the original timetable without any delay (Liebchen et al., 2009; Nielsen et al., 2012). Khosravi (2013) suggested two types of buffer times as running time buffer and dwell time buffer. Both are similar; the former adds extra time to the theoretical travel time to absorb delays from speed reduced regulations, called a running time buffer, while the latter adds extra time to the dwell time at the station to stop minor delays propagating. However, too large a buffer time will affect the optimal result because a decrease in the unnecessary gap will allow the company to operate more journeys and gain more benefits (Ushida et al., 2011). Therefore, the company should weigh-up optimisation and robustness of the timetable which directly connect with passenger reliability. The ideal is to find a balance for both the company and the passengers (Bertsimas et al., 2010).

2.8.4.2. Decreasing capacity

For capacity level, robustness can be increased by decreasing the theoretical maximum capacity level of physical infrastructure (Salido et al., 2008). This enables the timetable to cope with an unexpected event easier and still have some spare infrastructures which can be used for a minor delay or an infrastructure failure (Corman et al., 2011). For example, a spare platform or locomotive will resolve problems such as a broken engine or all platforms becoming unavailable. Planners can switch to use the spare infrastructures, and train operations can continue without any significant delay (Harrod, 2006). However, extra infrastructures are very expensive and may be viewed as luxury items by train companies. Therefore, there is a trade-off between cost of infrastructure and robustness of timetable (Schlechte et al., 2009). Companies search for alternative ways which can save cost and increase reliability.

2.8.4.3. Decreasing heterogeneity

Heterogeneity refers to the large rail traffic caused by trains that share the same infrastructure but have large differences in their characteristics of average speed and running duration (Hofman & Madsen, 2005). Freight and passenger trains also share the same infrastructures in large networks; this makes the timetable too complex and hard to control (Salido et al., 2008) and also results in delay through small headway times which spread to other trains in the same network (Salido et al., 2008; Vromans et al., 2006). By contrast, the term homogeneity is used to define trains that have similar characteristics and generate less train traffic (Hofman, 2005; Vromans et al., 2006). Both heterogeneity and homogeneity are directly related to the planning management stage and provide an impact that increases or decreases the robustness level of a timetable (Salido et al., 2008; Vromans et al., 2006).

2.9. Solution methods

This thesis focuses mainly on railway scheduling/rescheduling problems at the operational level of railway planning processes. Since the 1950s, scheduling problems such as machine scheduling and flow shop scheduling have been studied by many researchers (Potts et al., 2009). The first paper which discussed the RSP was by Szpigel (1973) and concerned the exact method to solve a small case problem. Since then, many have studied the basic RSP with a variety of constraints for network characteristics. However, the exact method is not efficient for solving a huge and complex RSP with limited computation time. An approximate method is required which can provide an acceptable solution in a reasonable time (Talbi, 2009).

2.9.1. Exact methods

The exact method guarantees the optimal solution for solving the optimisation problem: however, it usually requires extremely long computation time to solve complex, medium to large size problems (Talbi, 2009). Laporte et al. (1987) classified exact algorithms into three categories as (i) direct tree search methods such as branch and bound, branch and cut and branch and price, (ii) dynamic programming (DP) and (iii) integer programming (IP) and linear programming (LP) which are found in some commercial optimisation software such as CPLEX from ILOG and Xpress-MP from Dash (Talbi, 2009). Moreover, exact methods are usually used to solve the deterministic RSP.

Szpigel (1973) was the first to solve an optimisation on the railway scheduling problem using an LP model. They proposed a branch and bound method to reschedule and minimise total running time to deal with traffic or disruptions which occurred in the network. With this problem the transit positions, departure time and speed limit of trains were known but it was only tested on a small case study.

D'Ariano et al. (2007) studied a railway scheduling problem with real-time traffic management. They stated that when the disturbance occurred, new scheduling of the network should be re-computed within a limited time. They also developed a branch and bound method and alternative graph which they used to improve the computation speed. The experiment was based on a bottleneck area of the Dutch rail network. Results indicated that the branch and bound algorithm provided optimal or near optimal solutions within an acceptable time limit.

Caimi et al. (2012) proposed an integer programming model for rescheduling in a complex station area. They considered a predictive control approach and many alternative routes to help on dispatching decision-making with the objective of maximising customer satisfaction. For the computation, they implemented integer programming in IBM ILOG CPLEX Optimiser Version 12.3, which used data from the central railway station area Berne, Switzerland.

Donzella et al. (2018) solved a railway scheduling and rescheduling problem using LP formulations on Lingo software, aiming to reduce delays if a disturbance occurred and also minimise travel times of passengers to reach their destinations using alternative routing.

Their experiment used data from the railway network in Genoa, the capital of the Italian Liguria Region consisting of eight stations and considering scheduling and rescheduling of four different trains. Their model found the optimal solution within 30 seconds.

Zhang et al. (2019) studied the topic of integrated optimisation of train scheduling for sunset-departure and sunrise-arrival trains (SDS A-trains) and maintenance planning on the Beijing-Guangzhou high-speed line in China. SDSA-trains depart from the origin in the evening and arrive at the destination the next morning which attracts more passengers for long travel distance. However, regular maintenance on the high-speed railway in China is done at night and this impacts on SDSA-train scheduling. Therefore, they proposed MILP models with objective function to minimise the total travel time of SDSA-trains and reduce the effect of regular maintenance. They implemented the MILP model using CPLEX. Results showed that the model solved optimality in an acceptable time.

2.9.2. Approximate methods

Talbi (2009) stated that "Approximate methods generate high-quality solutions in a reasonable time for practical use but there is no guarantee of finding a global optimal solution". In real-life applications, the RSP is huge and complex because each set of data has their own characteristics based on their operation rule, regulation and infrastructures. The model was adapted depending on these constraints and this adds more complexity to the problem. Moreover, researchers should manage the whole railway network because disruption on one train will affect the whole system and the size of the problem is very large. Therefore, an approximate algorithm is normally applied to solve RSP. In general, approximate methods can be separated into two subclasses as heuristics and metaheuristics to solve the deterministic RSP. However, in the last two decades a simulation-based optimisation (sim-optimisation) has been applied to measure the realistic of using heuristics/metaheuristics to solve the stochastic RSP.

2.9.2.1. Heuristics

Talbi (2009) stated that heuristics usually find a good performance solution within acceptable time on large-size problem instances. Heuristics do not guarantee a feasible or optimal solution and are normally employed based on a rule of thumb or an educated guess. In addition, some heuristics are only designed to solve a specific optimisation problem. Some heuristics naturally relate to RSP such as local search (LS), Shifting Bottleneck (SB), Iterative Local Search (ILS) and Sequential Search Algorithm (SSA).

Brucker et al. (2005) proposed a LS to reschedule trains in cases where one track of a railway section consisting of two tracks in opposing directions was closed due to construction activities. Test results from instances of the German rail network showed that LS could be used to find a good and feasible solution.

The SB heuristic was used by Khosravi et al. (2012) to solve train scheduling and rescheduling problems. The problem was formulated as a job shop scheduling model using a UK railway network case study with the aim to reschedule and minimise the total delay of all trains. They decomposed the problem into several single-track railway scheduling problems. Results confirmed that SB found a feasible solution to all instances and was a good choice to solve this large and complex problem.

The ILS heuristic was provided to solve the railway scheduling and rescheduling problem for delay management in real-time railway traffic control (Corman et al., 2017). This paper mainly focused on the impact of rescheduling decisions on the quality of service perceived by the passengers. The experimental results were based on test cases of the real-world Dutch railway network and showed that this heuristic generated good quality solutions within a limited computation time.

Liu et al. (2018) implemented a Lagrangian relaxation-based approach heuristics and CPLEX to find a good solution of the train scheduling and train connection problem, combined with passenger flow control strategy. They set a model which proposed to trade-off utilisation of trains, passenger flow and number of passengers. Experimental results, based on three sets of instances using data from the Beijing-Yizhuang line, showed that Lagrangian relaxation-based approach had high performance compared with the CPLEX solver. Moreover, computation results also indicated that the proposed model was not sensitive to passenger demand.

Tian and Niu (2019) developed SSA to solve optimised train timetables with transfer synchronisation for a high-speed rail network. This study had a bi-objective function which aimed to (i) maximise the number of connections and (ii) minimise the total transfer waiting time. The case study was based on data from the Chinese High-Speed Railway Network which has a sub-network between Beijing and Guangzhou including four double-track lines, three transfer stations and eight non-transfer stations. Experimental results showed that SSA was effective in solving this problem.

2.9.2.2. Metaheuristics

According to Gendreau and Potvin (2010), metaheuristics can be defined as “high-level solution methods that provide guidelines to design and integrate subordinate heuristics to solve optimisation problems. These high-level methods characteristically focus on strategies to escape from local optima and perform a robust search of a solution space. Most of them are based, at least partially, on a neighbourhood search and the degree to which neighbourhoods are exploited varies according to the type of method.” Metaheuristics are approximate methods that do not guarantee to reach an optimal solution or know how near the solution is to the optimal solution. Metaheuristics provide an acceptable solution in a reasonable time for solving complex and large sized optimisation problems (Chica et al., 2017). Unlike heuristics, metaheuristics are general purpose algorithms which are easily applied to any optimisation problems (Talbi, 2009). Metaheuristics have been used in the RSP as Genetic Algorithms (GA), Tabu Search (TS), Simulated Annealing (SA) and Variable Neighbourhood Search (VNS).

A GA for the railway dispatching problem was presented by Ping et al. (2001). They experimented with a case based on the double track of Guangzhou to Shenzhen high-speed railway, with the aim of adjusting the railway scheduling and minimising the total delay of all trains. Results confirmed that the GA was efficient for this large size real-world problem in terms of computation time and solution.

Törnquist et al. (2005) applied two well-known metaheuristics algorithms as TS and SA separately to solve a railway traffic control problem that dealt with the uncertainty of disturbances. They formulated a MILP model to solve the RSP. This paper had a multi-objective to (i) minimise the total delay and (ii) minimise costs due to the types of delays. After experimentation, results confirmed that TS was outperformed by SA in the analysis of how each type of unpredictable event affected the total delay value.

Wang et al. (2015) studied a train scheduling problem for an urban rail transit network. They focused on three types of events as departure, arrival and change in passenger arrival rate. They also considered passenger transfer behaviour such as walking and transfer times. Their model aimed to trade-off between total passenger travel time and trains’ energy consumption. Results showed that both sequential quadratic programming (SQP) and GA were able to solve this problem. Moreover, SQP had better trade-off between performance and computational time than GA.

Samà et al. (2017) considered a train scheduling and routing problem using three types of solvers as the AGLIBRARY optimisation solver, TS and VNS. The problem was a railway traffic situation and they tried to improve the solution by rerouting or rescheduling with the objective function of the model to minimise train delays. The implementation started from a branch and bound algorithm to find an initial solution for the railway scheduling problem with fixed routes and then applied VNS and TS to solve train scheduling and routing problems for three case studies from Italy, the Netherlands and the United Kingdom. Results showed that VNS was outperformed when compared to TS and the AGLIBRARY optimisation solver.

Hassan and Reynolds (2018) proposed GA to schedule and optimise on the Roy Hill's Ore Train railway network, Western Australia. This paper aimed to minimise total running time of transporting iron ore to maximise revenue. GA provided better results compared to company's current solutions.

Nitisiri et al. (2019) implemented a parallel multi-objective GA with learning-based mutation to solve RSP. This paper aimed to increase the quality of service through two objective functions which were (i) minimise the average passenger waiting time and (ii) maximise the number of operating cycles. Two sets of experiments were based on a small case of the Bangkok Airport Rail Link (ARL) transit line in Thailand and a larger case of the Thailand Bangkok Rapid Transit System (BTS) transit network. Results showed that the proposed algorithm decreased average waiting time of the schedule and computation time.

2.9.2.3. Simulation-based optimisation (Sim-heuristics/metaheuristics)

Glover et al. (1996) introduced new methods which combined simulation and optimisation approaches together called Simulation-based optimisation (Sim-optimisation) to solve a real-world complex case study and measure the efficiency of the optimisation methods.

Figure 2.4 presents the overview picture of sim-optimisation. This algorithm has two parts which are stochastic nature and optimisation nature. Firstly, it works by using a simulation model to generate some random variables to the optimisation problem for simulating a real stochastic environment. The stochastic variable was set by using two steps as (i) choose a specific variable and (ii) define a probability distribution. Secondly, we used a heuristics or metaheuristics approach to find a near-optimal solution and then repeated these two phases until the end of the computation time limit. Therefore, the method provided n number of solutions to find an average of the objective function value (Glover et al., 1996; Glover et al., 1999). The increase in number of runs will increase reliability of a solution, however it also needs more computation time (April et al., 2003).

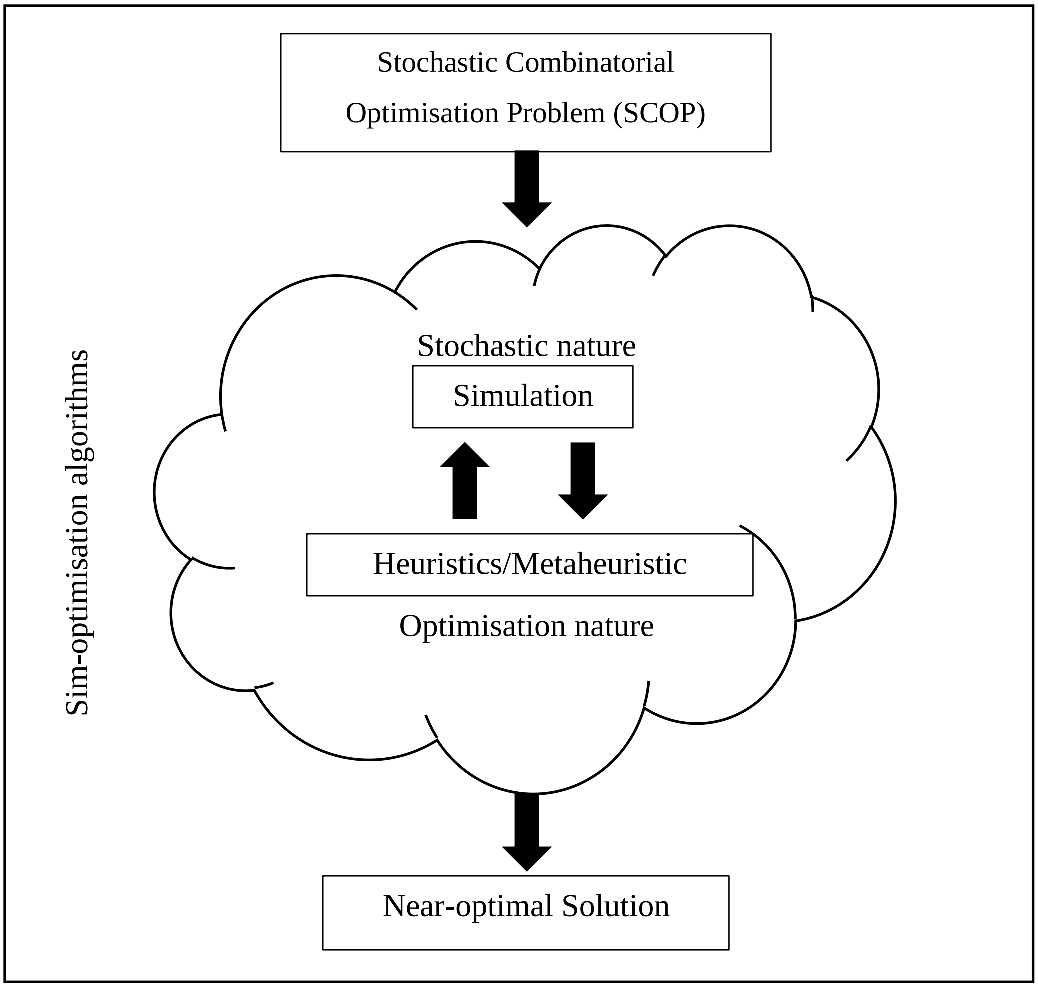

Figure 2.4 : Overview scheme of Sim-optimisation approach (Juan et al., 2015)

These methods have been used successfully in other approaches such as manufacturing problem (Chong et al., 2003; Klemmt et al., 2009; Frantzén et al., 2011), flow shop scheduling problem (Yang et al., 2004), and job shop scheduling problem (Arakawa et al., 2003; Nicoară et al., 2011; Korytkowski et al., 2013).

To the best of our knowledge, Sajedinejad et al. (2011) is only one who used a sim-optimisation to solve a railway problem. They focused on finding a near-optimal train timetable using Enterprise Dynamics (ED) as commercial simulation software and GA to minimise total train running time. A large-scale network based on capacity, train and infrastructure characteristics was used as a case study. Hassannayebi et al. (2014) also proposed a sim-optimisation method on GA to solve the railway transit problem. The ED was used to simulate a stochastic number of passengers, running time and dwell time. This paper aimed to minimise the average passenger dwell time at the stations by considering uncertainties.

2.10. Chapter Conclusion

This chapter provided a railway planning process as RSP under the uncertainties of topology of network, signalling system, disruption management, how to handle the RSP and solution methods of RSP. The signalling system helped to identify characteristics of the problem when creating the model for solving the RSP.

Generally, the RSPs are usually classified into two main types based on their input data as the deterministic RSP and the stochastic RSP. Basic knowledge was provided on the topology of network, classification of disruption, impact of disruption and ways to handle them. All these data can help to present an overview of disruption management.

Then, a review of solution methods for solving the RSP by various types of algorithm was discussed using exacts, heuristics, metaheuristics and sim-optimisation methods. Therefore, all knowledge from the literature reviews will be applied and used in the next chapter of the thesis to solve RSP with deterministic and stochastic of uncertainties delay.

Part I

Deterministic Railway Scheduling in the presence of uncertainties

Chapter 3: Iterated Greedy with Biased Randomised (IG-BR) for deterministic RSP

3.1. Introduction

In the deterministic RSP, all disruptions which we considered such as running time, dwell time and late departure time delay should be known in advance as the input data before the start to solve the problem, this can be called as deterministic delays. The deterministic RSP is a large and complex combinatorial optimisation problem (COP) because it deals with complex infrastructures, large amounts of data and decision variables which directly affect the difficulty of decision-making by management. An optimisation model is defined to consider all characteristics of the railway network, which are defined in terms of objective function, decision variable and constraints mainly comprising headway and signalling. The headway constraints are formulated for both following and opposite trains, and signalling constraints are only considered for the consecutive trains. Signalling constraints use to managing the necessary buffer time between two following train depend on the network signalling system of each country. Due to the complexity of the RSP, it is very hard to find optimal solutions for the problems, therefore this study intends to formulate RSP in order to use the heuristics and metaheuristics algorithm to solve the problem.

The IG algorithm is a heuristic that was introduced to solve the permutation flowshop scheduling problem to minimise the makespan. Results showed that IG was very effective in providing a new best solution (Ruiz et al., 2007). IG algorithms have been used successfully to solve many COPs; however, the RSP can be considered as a job shop scheduling problem or machine scheduling problem. According to safety regulation, each block can operate only one train at a time which is similar to a job shop scheduling problem where only one job can be processed at a time on a machine. Using this analogy, trains are presented for jobs and blocks are considered as machines. Furthermore, Ying et al. (2010) proposed an IG algorithm to solve a parallel machine scheduling problem considered with a dependent setup time. Results revealed that the IG heuristic provided a high performance when compared to the literature by using the same benchmark problem. Moreover, Bresina (1996) and Cáceres-Cruz (2013) claimed that biased randomisation (BR) can help to improve the solution of any heuristics by performing a diversified exploration of the solution space. Therefore, in this chapter, we propose an IG-BR algorithm to solve the RSP that deals with a deterministic delay.

In this chapter, the basic concept of IG and BR algorithm is discussed as a heuristic method that is widely studied to improve the efficiency of COPs. A method to use IG-BR to solve a deterministic RSP is also presented. Explanations of each case study are presented in detail. Data were collected from the UK and Thailand rail networks. Experimental results showed the efficiency of the proposed methods and how they improved a real-life case study with short computation time.

The chapter is structured as follows: Section 3.2 presents the chapter contributions and the deterministic RSP optimisation model is proposed ln section 3.3. Section 3.4 presents the proposed IG with the BR algorithm to solve RSP with deterministic disruption delays. Data collection, test instances and all the experiments results are presented in section 3.5. Results are separated into two case studies (UK & Thailand) while section 3.6 concludes this Chapter.

3.2. Contribution

In the literature, the IG algorithm has been used successfully and effectively to solve many OR problems such as Task assignment (Harish et al., 2014), Task allocation (Kang et al., 2013), Parallel machine scheduling (Ying et al., 2010), Freight train scheduling (Yuan et al., 2008), Non-permutation flow shop scheduling (Ying, 2008), and Flow shop scheduling (Ruiz et al., 2008). Since IG has been adapted successfully for many COPs, it is suitable to combine with other techniques to improve the solution of the IG algorithm and the BR technique was selected. The BR with some heuristics was first introduced by Bresina (1996) who used Heuristic Biased Stochastic Sampling as a search technique for a scheduling problem which outperformed in a small size case study. Moreover, Cáceres-Cruz (2013) claimed that the BR technique can add diversity to the main algorithm.

Other reasons behind using the IG-BR heuristic to solve deterministic RSP were that the IG algorithm is easy to implement, parameter free and easy to adapt to different sizes and types of problem. Furthermore, the BR algorithm is not a complex process and is fast in terms of computation time, easy to combine with every algorithm and can provide randomised behaviour for the search space.

To the best of our knowledge, the combination of IG and BR has not been used in the RSP before; therefore, this is the first study to use biased randomised to improve the IG algorithm for the RSP dealing with delays.

3.3. Deterministic RSP optimisation model

The RSP optimisation model has been modelling tree search graph model to describe the characteristics of the rail network and used it to develop an optimisation model to solve the deterministic RSP.

3.3.1. Tree search graph

Roy and Sussmann (1964) introduced the disjunctive graph to solve the scheduling problem. They proposed the concept of conjunctive graph and disjunctive graph to model the job shop scheduling problem for minimising the makespan. Furthermore, the concept of an alternative graph was provided by Mascis et al. (2002) to solve a job shop scheduling problem by adding extra arcs called alternative arcs to consider blocking and no-waiting time constraints. However, in this thesis, we used a tree search graph to model a RSP by using the longest path search to find the delay of each train. Figure 3.1 below shows an overview of the basic tree search structure.

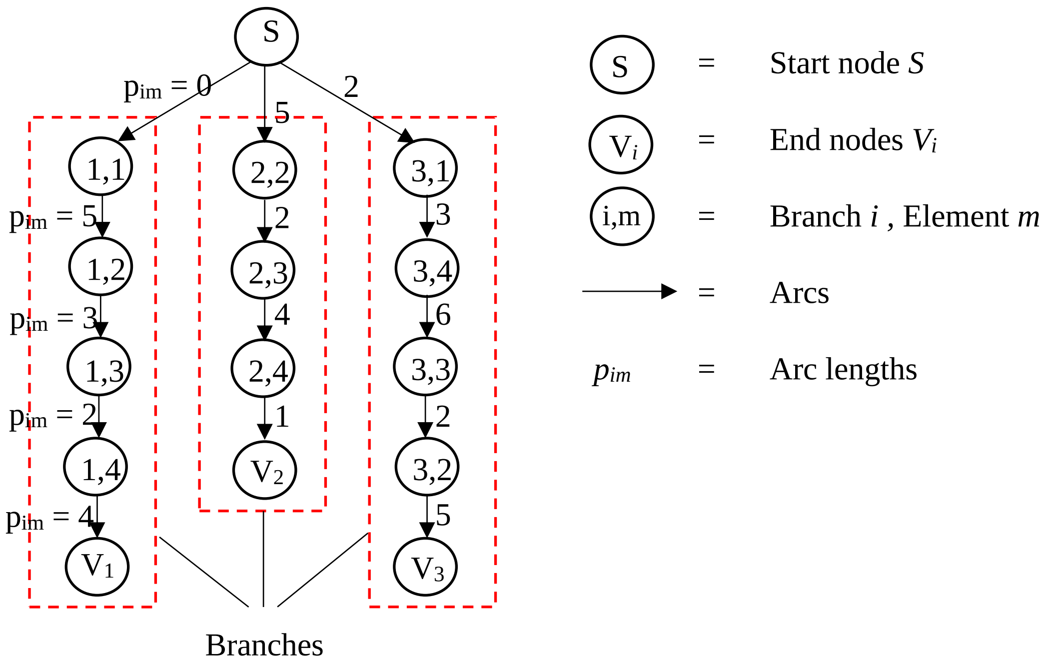

Figure 3.1 : Tree search graph structure

The graph $G = (N,A)$ is described with a set of nodes shown by N set of conjunctive arcs A. In Figure 3.1, a value is shown by a node (i,m) in set N, where i refers to number of branches and m refers to number of elements. Set N has two types of nodes as actual and virtual. Virtual nodes contain two special nodes, namely a start node S and an end nodes V_i. The set of arcs A uses a solid line with one direction arcs that refers to the direction between all elements of branches i. For example, a line between (i,m) and (i,n) with the direction point to *(i,n)* means in branches *i*, an element *m* starts before the element *n*. Moreover, values at the arcs p_{im} are equal to arc lengths between two elements in the same branch. Finally, the number of arcs that connect from the start node show as the number of branches in this figure, indicated by a rectangular red dashed line.

In addition, we compared characteristics and structure between the tree search graph (Figure 3.1) and railway scheduling problem as shown in Table 3.1 as follows:

Tree Search	**Railway scheduling**
Nodes (Branches and Elements)	Operation (Trains and Blocks)
Arcs	Blocks
Arc lengths	Running/Dwell time and Departure from origin
Branch	Route

Table 3.1 : Comparative structure between Tree Search and Railway scheduling

3.3.1.1. Conjunctive arcs

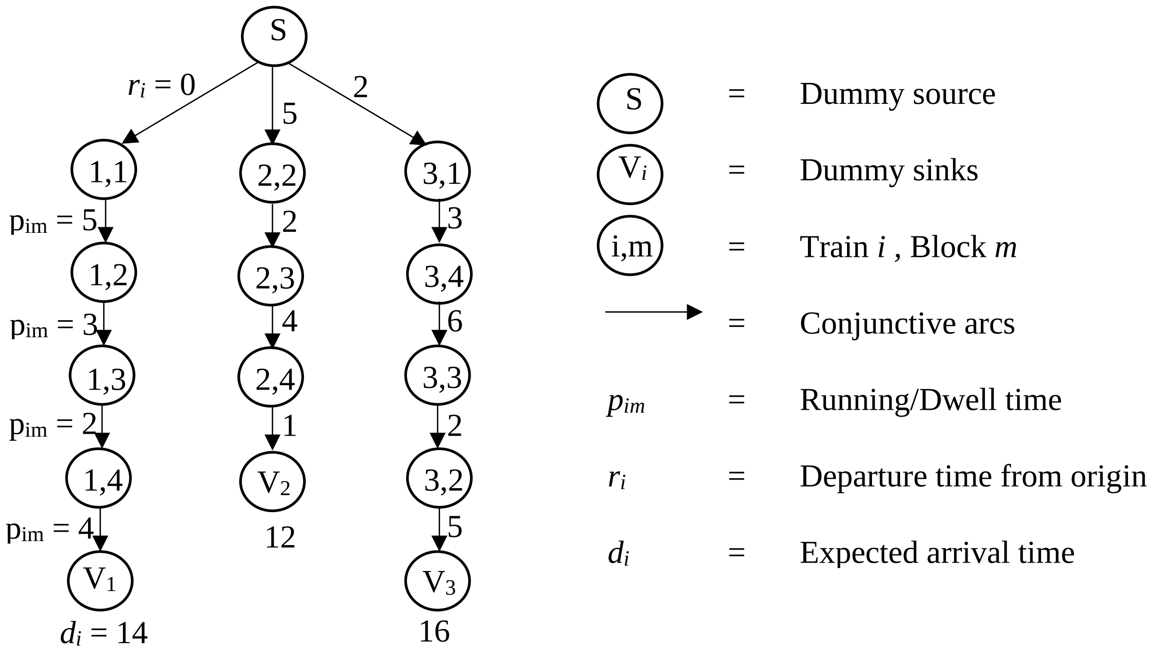

Figure 3.2 : Conjunctive graph

The graph $G = (N,A)$ is described with a set of nodes shown by *N* set of conjunctive arcs *A*. In a Figure 3.2, an operation is shown by a node *(i,m)* in set *N*, where *i* refers to number of trains and *m* refers to number of blocks. Set *N* has two types of nodes as actual and virtual nodes. In virtual nodes there are two special nodes, namely a dummy source *S* and dummy sinks V_i. The set of conjunctive arcs *A* uses a solid line with one direction arcs which refers to the direction and route sequence between all blocks of train *i*. For example, a conjunctive line between *(i,m)* and *(i,n)* with the direction point to *(i,n)* means train *i* should be travelling from *m* to *n*.

Moreover, the values of the length of the conjunctive arcs are equal to the running/dwell time p_{im} of the train between two stations. However, arcs which connect the dummy source node *S* to the first operation of each train route is different; the length of these arcs can be used to identify the departure time r_i of train *i*. On the RSP, the value of departure time r_i should be fixed, because all trains need to travel after the departure time in a schedule.

At the dummy sinks V_i, the expected arrival time d_i has been set and fixed in the schedule. If we sum up the departure time r_i and running/dwell time p_{im} of the train *i*, this formulation provides an actual arrival time C_i of train *i* to its destination. Then, we considered if the train *i* has an actual arrival time C_i before the scheduled arrival time d_i to the destination. If not, it means that the train was delayed and has a train delay T_i. Therefore, Equation 3.1 below shows how to calculate the delay of each train *i*.

$$T_i = \max(C_i - d_i\,,\ 0) \tag{3.1}$$

3.3.1.2. Disjunctive arcs

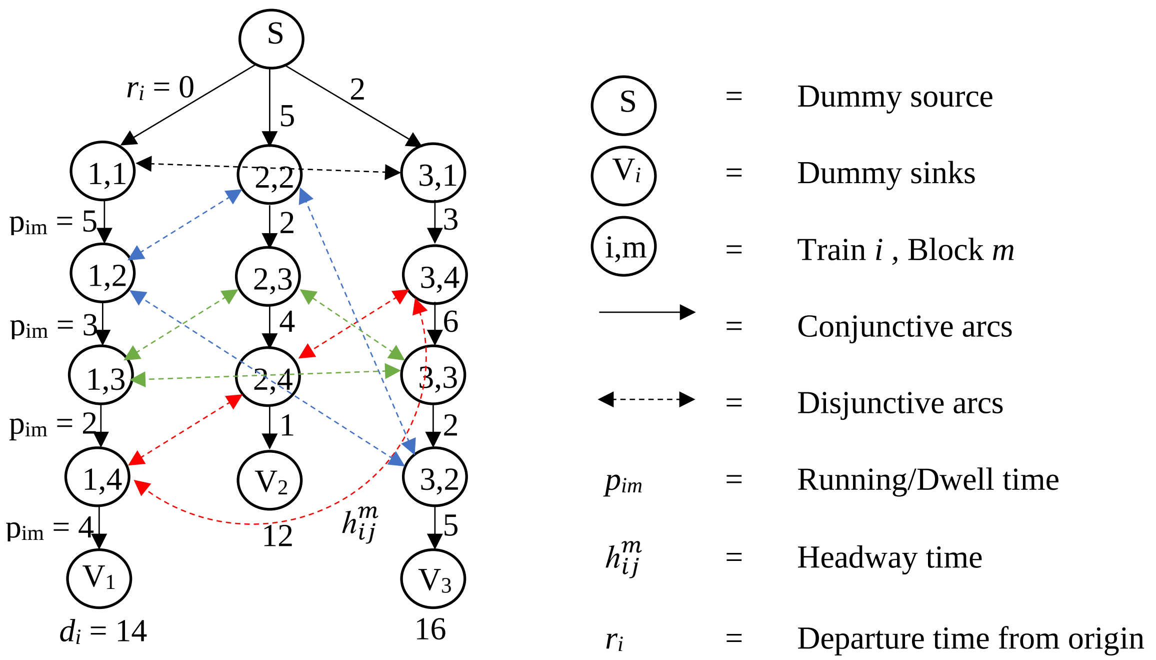

Figure 3.3 : Conjunctive graph and Disjunctive graph

In Figure 3.3, the disjunctive arcs have been added into the previous conjunctive graph model (Figure 3.2). Pinedo et al. (1999) introduced the disjunctive graph to solve a job shop scheduling problem with jobs that have different release times. Therefore, this thesis adapted the disjunctive graph concept to add the headway constraints into the basic RSP model. The headway constraint refers to a dwell time between two consecutive trains (i, j) that operate in the same block m, this time is called headway time (h_{ij}^{m}). On Figure 3.3, the graph *G = (N,A,B)* is described each operation *(i,m)* represented by *N*, set of conjunctive arcs *A*, dummy source node *S*, dummy sinks node V_i, running/dwell time p_{im}, and departure time r_i are the same. A set of disjunctive arcs *B* are added by using a dash line indicating bidirectional arcs referring to the headway constraints (h_{ij}^{m}), and the path choice. For example, the disjunctive arcs between *(i,m)* and *(j,m)*, the bidirectional arcs which means that the model needs to choose one direction for the bidirectional arcs to fix the order of two trains *i* and *j*. If we choose the direction to *(j,m)*, this means train *i* has to operate before train *j* in the block *m*. Otherwise, train *j* has to operate before train *i* in the block *m*. In this RSP case study, the values at the length of the disjunctive arcs are shown in the headway (h_{ij}^{m}) between two consecutive trains which operate in the same block *m.*

To have a clear view of the disjunctive arcs, we drawn them in a different colour for each station. The delay of each train has been calculated the same as the section 3.3.1.1 (equation 3.1); however, we need to consider both the conjunctive and disjunctive arcs by using longest path calculations. The departure time of the train to the next station r_{im} was calculated by finding a maximum between the sum of all previous running/dwell time in their own route of train *i* (conjunctive graph), and a sum of all previous running/dwell time and headway time of the corresponding train *j* (disjunctive graph).

3.3.1.3. Alternative arcs

The alternative graph was first introduced to solve the scheduling by Mascis et al. (2002). In this thesis, we used alternative arcs to model train scheduling. A clear picture of applied alternative graph for RSP is shown in Figure 3.4 (a) two-aspect signalling system (D'Ariano et al., 2007) and Figure 3.4 (b) four-aspect signalling system (Khosravi et al., 2012). A pair of alternative arcs is applied to tree search model when two trains *i* and *j* need to operate on the same block *m*, they need to wait for the green signal from the other train. If *(i,m)* and *(j,m)* are two trains which require to enter the same block, we set *(i,* $S_i(m)$*)* and *(j,* $S_j(m)$*)* as their set of next successor operation in a sequence of train *i* or *j* for two-aspect signalling and set of next three successor operations in a sequence of train i or j for four-aspect signalling. In addition, the next successor operation or next three successor operations refer to the next block or next three blocks that will be traversed by train i or j. In this pair of alternative arcs, we need to choose only one arc from node *(i,* $S_i(m)$*)* to *(j,m)* or the arc from node *(j,* $S_j(m)$*)* to *(i,m)*.

The way to select one of the alternative arcs is dependent on the disjunctive arcs that we choose. Figure 3.4 shows that if we choose disjunctive arc from to *(i,m) to (j,m)*, that means we also have the alternative arc from node *(i,* $S_i(m)$*)* to *(j,m)*, otherwise select the alternative arc from node *(j,* $S_j(m)$*)* to *(i,m)*. For example, the set of next operation $S_i(m)$ for two-aspect signalling is shown in Figure 3.4(a) as $m = \{2\}$ and the set of the next three operations $S_i(m)$ for four-aspect signalling are shown in Figure 3.4(b) as $m = \{2, 3, 4\}$. Moreover, the length of all alternative arcs is equal to zero, because it is only used for making node connections to each other and for adding signalling constraints to a model.

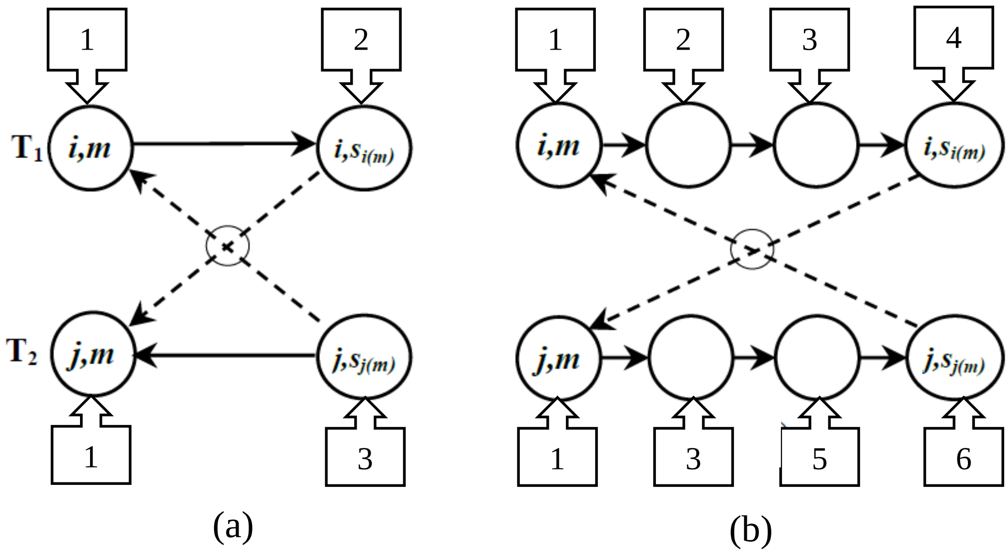

Figure 3.4 : Alternative graph (a) two-aspect signalling and (a) four-aspect signalling (Khosravi, 2013)

For the calculation example, if we scheduled operation *(i,m)* before operation *(j,m)*, we need to sum up a running/dwell time of the current operation *(i,m)*, and next operations in a sequence of train i for two-aspect signalling or sum up until the next three operations in a sequence of train i for four-aspect signalling ($\sum_{(i,k)\in S_i(m)} p_{ik}$). Therefore, the operation *(j,m)* can only wait and start after the signalling constraints ($\sum_{(i,k)\in S_i(m)} p_{ik}$), which means the next block is free to enter, in other words the signal light should be turned to green.

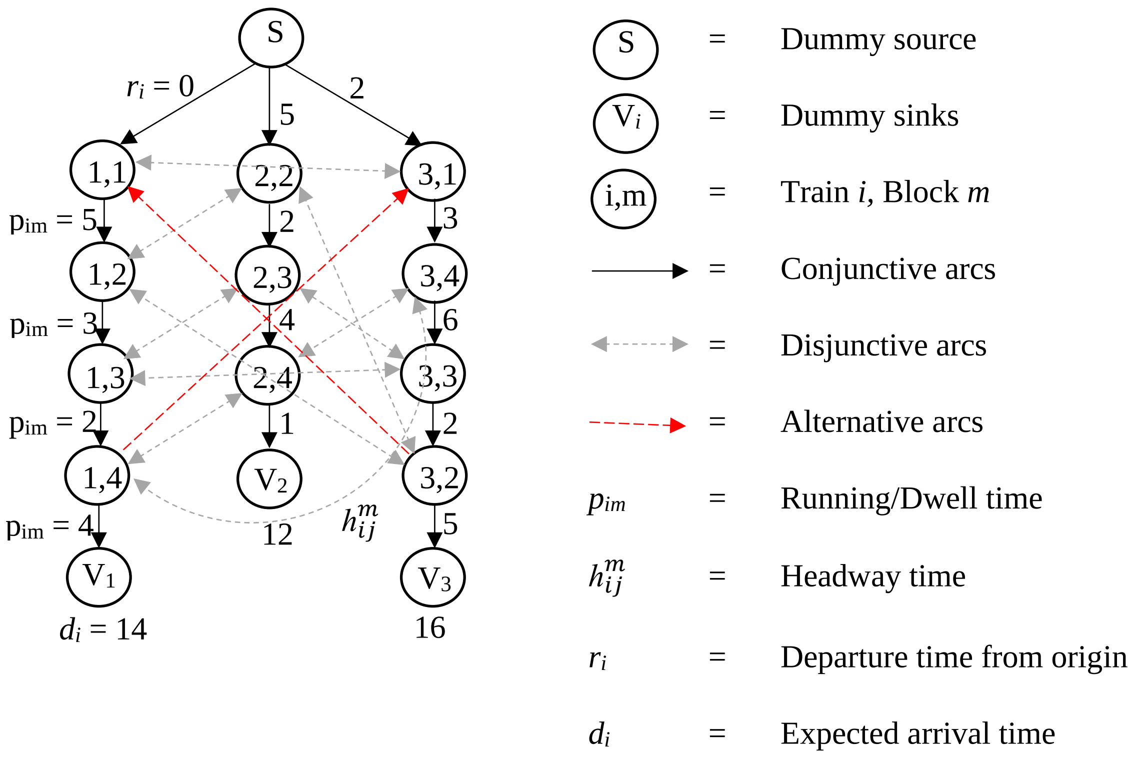

Figure 3.5 : Conjunctive graph, Disjunctive graph and Alternative graph for four-aspect signals

In Figure 3.5, alternative arcs have been added into the conjunctive graph, and disjunctive graph model (Figure 3.3). The graph *G = (N,A,B,C)* is described, each operation *(i,m)* is represented by *N*, set of conjunctive arcs *A*, set of disjunctive arcs *B*, dummy source node *S*, dummy sinks node V_i, running/dwell time p_{im}, departure time r_i, and headway time h_{ij}^m are the same. In our RSP case study, a train needs to wait at the station until the signalling turns green. Therefore, the set of alternative arcs *C* is added by using a long red dashed line with one direction arcs referring to signalling constraints and path choice. For example, if we order train 1 before train 3 at station 1, train 3 should be waiting and depart from station 1 after train 1 arrives at station 4 or operation node (1,4). This can be calculated by the sum of running/dwell time p_{im} of train 1 from station 1 to 4 (5 + 3 + 2 = 10); a clearer picture of alternative arcs is shown in Figure 3.5.

Moreover, we also studied about how to develop the RSP model by using the Tree search graph method which includes all objective functions and constraints, especially headway and signalling constraints.

3.3.2. Optimisation model

In the deterministic RSP optimisation model, we considered the disruptions which are (i) departure time delay and (ii) running time and dwell time delay. Both types of delay were added into the parameters below as the input data before solving the problem:

r_i	departure time of train i from its origin
$p_{i,m_{i,k}}$	indicate the running time of train i on the block m at the position k, for $k = 2,\ 3,\ \ldots,\ l_i$

Table 3.2 : List of parameters for adding the uncertainties delay value

We added the departure time delay into expected departure time r_i, then added the running time and dwell time delay into the running time of train i on the block m at the position k $p_{i,m_{i,k}}$.

The following table shown more parameters and decision variables of the general optimisation model for the RSP.

I	set of trains
M	set of blocks denoted by $M = \{1,\ 2, \ldots\ , m\}$
i, j	indicate for trains $i = \{1,\ 2, \ldots\ , I\}$ and $j = \{1,\ 2, \ldots\ , J\}$
d_i	scheduled arrival time of train i at its destination
w_i	importance weight of train i
$t_{i,m_{i,k}}$	indicate the depart time of train i on the block m at the position k, for $k = 2,\ 3,\ \ldots,\ l_i$, where l_i is the last block of sequence
g_{im}	indicate the depart time of train i on block m
p_{im}	running/dwell time for train i on block m
h_{ij}^m	required time delay (headway) between two consecutive trains (i, j) that operate in the same block m
$(m_{i1},\ m_{i2},\ \ldots,\ m_{i,l_i})$	sequence of blocks (routes) to be traversed by train i, where l_i is the last block of sequence
$(i,\ m)$	Train and block indices, for $i\ \in I$, and $m\ \in M$
O	set of operations defined by indices $(i,\ m)$, for $i\ \in I$, and $m\ \in M$
$s_i(m)$	the immediate successor block (the third successor block) of $(i,\ m)$ for two-aspect signalling (four-aspect signalling)

$S_i(m)$	a set containing train and block indices $(i,\ m)$ for two-aspect signalling, additionally containing the indices of its immediate and second successor operations for four-aspect signalling
$n_{is_i(m)}$	indicate the depart time of train i on immediate successor block (the third successor block) of $(i,\ m)$ for two-aspect signalling (four-aspect signalling)
T_i	delay of train i which occurred when the actual arrival time C_i of train is greater than its expected arrival time d_i
x_{ij}^m	binary variable to determine the order of train that operate on the same block, where x_{ij}^m equal to 1 when train i operate on block m before train j, otherwise x_{ij}^m equal to 0

Table 3.3 : List of parameters and decision variables in the mathematical model formulation

The RSP optimisation model was adapted from the classical job shop scheduling model with some additional constraints, and called this Modified Blocking Job Shop Scheduling (MB-JSS) model (Khosravi, 2013), which was also developed from the tree graph search model. In this problem, we assume that a set I is the set of trains which are operated on a set of blocks m denoted by $M = \{1, 2, \ldots, m\}$. Moreover, i and j are used to indicate for each train, where $i = \{1, 2, \ldots, I\}$ and $j = \{1, 2, \ldots, J\}$. Each train i should travel in the specific order as $(m_{i1},\ m_{i2}, \ldots,\ m_{i,l_i})$, where l_i is the last block of sequence. When each train starts on the block it must finish without any interruption (no pre-emption is allowed). Each train has its departure time r_i. All trains can start after the departure time and have a fixed running/dwell time p_{im} to the next block, where (i, m) indicates to train and block, for $i\ \in I$, and $m\ \in M$. In addition, one block can only operate one train at a same time.

The objective is to find the sequence of train on each machine which provides a minimised total weighted of train delays $\sum w_i T_i$, let T_i be a train delay which occurred when the actual arrival time C_i of any train is greater than its expected arrival time d_i, mean the train delay was occurred, therefore the train delay was calculated by $T_i = \max(C_i - d_i\ ,\ 0)$, where w_i is the priority of each train. The actual arrival time C_i was calculated from the depart time of train i on the block m at the last block of sequence l_i plus the running time at that position, which is $t_{i,m_{i,l_i}} + p_{i,m_{i,l_i}}$. Moreover, we set $t_{i,m_{i,k}}$ to indicate the depart time of train i on the block m at the position k, for $k = 2, 3, \ldots, l_i$, and $p_{i,m_{i,k}}$ to indicate the running/dwell time of train i on the block m at the position k, for $k = 2, 3, \ldots, l_i$.

According to the rail network regulations, we also need to consider the safety reasons which are headway and signalling. The headway is the required dwell time between two consecutive trains that operate on the same block h_{ij}^{m}, where i, j is used to indicate the number of trains, and m is used to indicate the number of blocks. Then, the train needs to stay at the previous block until the next block is available (green signal is shown). In the UK, we have used four-aspect signalling system which considers that the next four blocks should be free, so let $s_i(m)$ be the third successor block. However, in Thailand, we have considered the two-aspect signalling system which considers that the next two blocks should be free, so let $s_i(m)$ be the next successor block. $S_i(m)$ a set of containing blocks for two-aspect signalling or four-aspect signalling. In addition, the signalling constraints are depended on the characteristics of safety regulations in each country. This method can be adapted to use for all countries by changing the number of blocks in a set of containing blocks $S_i(m)$. Moreover, a block can be occupied by only one train at a time according to line blocking which is a safety regulation for train management.

Furthermore, we set x_{ij}^{m} as a binary variable to determine the order of trains that operate on the same block, where x_{ij}^{m} is equal to 1 if train i operates on block m before train j, otherwise x_{ij}^{m} is equal to 0. Then, let $\mathcal{O}$ be the set of operations indicated by (i, m), for $i \in I$, and $m \in M$, and let $\mathcal{N}$ be the large amount of positive number.

Generally, in order to minimise the total weighted delay of the train, a mixed integer linear programming (MILP) model is developed to solve the problem as following:

$$Minimize \quad z = \sum_{i \in I} w_i T_i \tag{3.2}$$

Subject to:

$$T_i \geq t_{i,m_{i,l_i}} + p_{i,m_{i,l_i}} - d_i \qquad i \in I \tag{3.3}$$

$$t_{i,m_{i,1}} \geq r_i \qquad i \in I \tag{3.4}$$

$$t_{i,m_{i,k}} \geq p_{i,m_{i,k-1}} + t_{i,m_{i,k-1}} \qquad i \in I; \quad k = 2, 3, \ldots, l_i \tag{3.5}$$

$$g_{jm} - t_{im} + \mathcal{N}\left(1 - x_{ij}^{m}\right) \geq \max\{p_{im}, h_{ij}^{m}\} \qquad (i, m), (j, m) \in \mathcal{O} \tag{3.6}$$

$$g_{im} - t_{jm} + \mathcal{N}\left(1 - x_{ji}^{m}\right) \geq \max\{p_{jm}, h_{ji}^{m}\} \qquad (i, m), (j, m) \in \mathcal{O} \tag{3.7}$$

$$g_{jm} - n_{is_i(m)} + \mathcal{N}\left(1 - x_{ij}^{m}\right) \geq \sum_{(i,k)\in S_i(m)} p_{ik} \qquad (i,m),(j,m) \in \mathcal{O} \tag{3.8}$$

$$g_{im} - n_{js_j(m)} + \mathcal{N}\left(1 - x_{ji}^{m}\right) \geq \sum_{(j,k)\in S_j(m)} p_{jk} \qquad (i,m),(j,m) \in \mathcal{O} \tag{3.9}$$

$$x_{ij}^{m} + x_{ji}^{m} = 1 \qquad (i,m),(j,m) \in \mathcal{O} \tag{3.10}$$

$$x_{ij}^{m} \in \{0,1\} \qquad (i,m),(j,m) \in \mathcal{O} \tag{3.11}$$

$$T_i \geq 0 \qquad i \in I \tag{3.12}$$

According to above optimisation model, $(equation\ 3.2)$ is the objective function which minimises the total weighted of train delays. The train delay of each train is calculated by constraints $(equation\ 3.3)$ as the actual arrival time minus the expected arrival time. Constraints $(equation\ 3.4)$ ensure that all trains cannot start before their departure time.

The conjunctive constraints are shown in $(equation\ 3.5)$, which set that each train operates in the specific order sequence; the departure time of each train on each block should be calculated from the departure time of previous block plus the running/dwell time.

The disjunctive constraints are defined in $(equation\ 3.6)$ and $(equation\ 3.7)$, these formulas identify the sequence of the different trains in the same block and adapt to provide a minimum headway between two consecutive trains.

The alternative constraints $(equation\ 3.8)$ and $(equation\ 3.9)$ are adapted to define the signalling system. All trains should wait at the previous block until the next block available.

In $(equation\ 3.10)$, the constraints set that the train can only appear once on each block. Decision variable is defined in $(equation\ 3.11)$ as the binary number, and equation $(equation\ 3.12)$ defines that delay should be a positive value.

In addition, the difference between UK and Thailand optimisation models are shown in $(equations\ 3.6 - 3.9)$, which are headway constraint and signalling system constraint. For the minimum headway between two consecutive trains in $(equations\ 3.6 - 3.7)$, the UK regulation has 150 seconds of headway time and Thailand has no headway time between two consecutive trains. Therefore, in Thailand optimisation model we can delete this constraint out from the model. Moreover, on the signalling system constraint in ($equations$ $3.8-3.9$), UK rail networks run based on four-aspect signalling and Thailand runs following two-aspect signalling, so we can select which types of signal that we need to consider by changing the number of a set of containing blocks $S_i(m)$ as three blocks for four-aspect signalling (UK) or one block for two-aspect signalling (Thailand).

3.4. Proposed IG with Biased Randomised (IG-BR) for solving RSP

The large and complex problems are very hard to solve by using the exact method, because it required very huge computation time to solve a problem. Then, many methodologies based on heuristics and metaheuristics were developed to deal with large-scale COPs to support the decision-making management. Therefore, we have selected the Iterated Greedy (IG) algorithm which is one of the well-known heuristics that very successfully and effectively to solve many OR problems to solve the RSP.

3.4.1. Basic concept of IG algorithm

The basic concept of IG algorithm is shown in Figure 3.6. In this figure, it presents that the IG algorithm process has been separated into two phases which are the destruction phase, and construction phase. This example shows the IG process of the sequence solution for all trains in one block.

The IG algorithm starts on the destruction phase by choosing trains randomly from the initial sequence and keeping those trains in the construction set, then using trains which have been selected out to reinsert in the construction process. Therefore, the initial sequence is separated into two sets of trains which are the remaining trains set and the construction set.

Then, in the construction phase process, we reinsert the first train in the construction set back into all possible positions of the remaining trains set. In each inserted position, the objective value should be calculated by sum up the delay of each train together to find the minimise total train delay T_{min}. Then, we select the sequence which provides the best value as the remain trains set. Therefore, these steps are repeated until the destruction set is empty and this process counts as one iteration (Ribas et al., 2011; Ruiz et al., 2008; Ruiz et al., 2007).

According to the structure of the IG algorithm, all processes is working based on a parameter free which mean this algorithm are easy to implement, and adapt to use with many COPs (Ruiz et al., 2007).

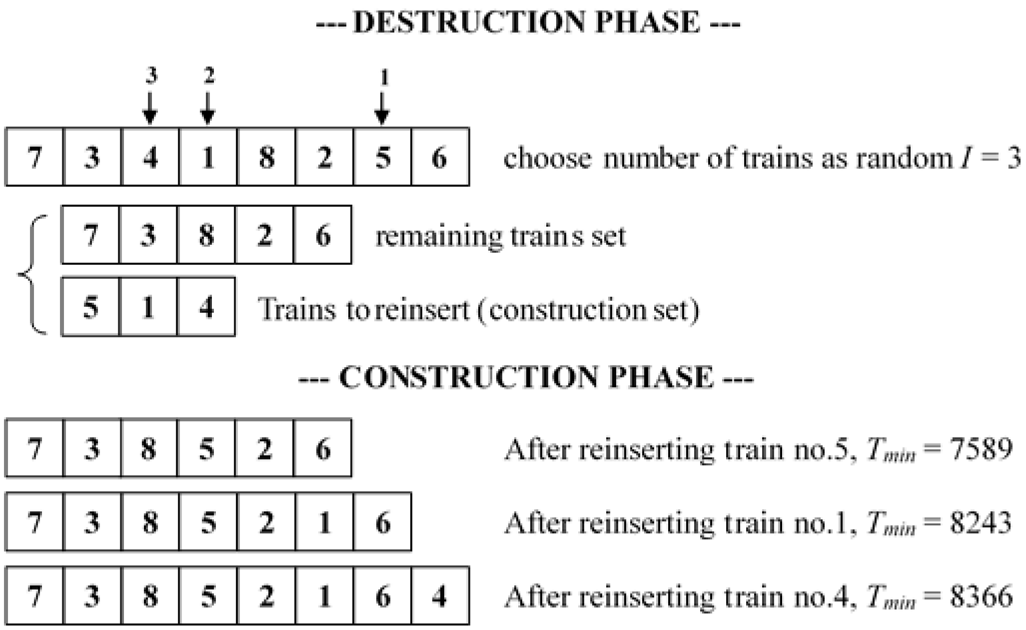

Figure 3.6 : Example for one iteration of the IG algorithm (Ruiz et al., 2007)

Algorithm 3.1 presents the IG Algorithm of the RSP, which shows how to use basic IG to solve the railway scheduling problem. Firstly, the process starts by creating an initial solution based on the given timetable (Pseudocode: line 2). Then, we start to implement the IG for every block of the rail network to reorder the sequence (Pseudocode: line 5). In each block, we need to do the destruction phase and construction phase.

In the destruction phase of IG, we randomly select a few trains from the initial sequence and put them in the construction set, so the trains left in the sequence are called the remaining set (Pseudocode: line 5 to line 8).

Next, in the construction phase, we reinsert the construction set back into the remaining set by trying all possible positions and finding the best sequence that provides the best solution (Pseudocode: line 9 to line 14). Then, we need to check that operational constraints of the best sequence such as departure time, running/dwell time and headway constraints are still valid (Pseudocode: line 11 to line 13). Finally, the new schedule is generated to manage the rail network (Pseudocode: line 15).

Algorithm 3.1 : Procedure for RSP IG Algorithm

Procedure for RSP IG Algorithm (r', r^D, r^R, d, b)
▷ r': Initial Solution
▷ r^D: Partial sequence to reconstruct
▷ r^R: Trains to reinsert
▷ d: Random chosen number of trains
▷ b: number of blocks
r': *GenerateInitialSolution*(); ▷ *Initial Solution*
set r^R = empty
for i = 1 to b **do** ▷ *run for all block*
  for i = 1 to d **do** ▷ *Destruction step*
    r^R <- remove one node randomly from r' and insert it in r^R;
    $r^D = r'$ <- the remaining set of trains
  end for
  for j = 1 to d **do** ▷ *Construction step*
    r^{best} = best solution obtained after inserting train from r^R in all possible position of r^D
    if r^{best} met with all railway constraints **then** ▷ *check all constraints*
      solution (r^{best}) = calculate minimise a total delay (r^{best})
    end if
  end for
solution(r^{best}) <- Using the best sequence r^{best} in each block to generate a timetable
end for
end procedure

3.4.2. IG with Biased Randomised (IG-BR)

In this section of the research, we add the BR technique to improve the quality of the solution from the basic IG algorithm. The pseudo-code of the basic IG algorithm is presented as Algorithm 3.1.

To the best of our knowledge, Bresina (1996) is the first paper which used the BR heuristic studied. This paper presented a Heuristic-Biased Stochastic Sampling (HBSS) method to solve a scheduling problem. The results of this approach provided a better solution than a greedy search in a small sample.

Juan et al. (2013) developed multi-start BR of heuristics (MIRHA) with a LS for solving non-smooth vehicle routing problems (VRPs) by adding a biased random behaviour for the LS algorithm. Moreover, they suggested that the concept of applying biased probability distributions to randomise classical heuristics was an effective method for many COPs.

Juan et al. (2015) developed BR with ILS heuristic to solve the multi-depot VRP. This method applied BR with geometric probability distribution for generating a diversification for the ILS and experiment on standard benchmarks. Results confirmed that BR with ILS outperformed as the best result in the literature.

Agustin et al. (2016) used BR for solving the Crew Pairing Problem (CPP). The experimentation implemented a multi-start algorithm based on BR. The result from computational experimentation gave an effective solution.

The literature review illustrated that BR improved the performance of algorithms by avoid the local optimal, increase diversification of the heuristics or metaheuristics and resulted in high quality solutions. Thus, it is reasonable to combine BR techniques with IG heuristic to solve RSP.

We extend the basic IG algorithm by adding the BR concept to improve the quality of the solution. In the construction phase of the IG heuristics, we selected the first element in the construction set to reinsert it in the sequence and find the best sequence that gives the best objective function value. However, in the basic IG, we used the uniform distribution for the randomisation of that list which basically destroyed the logic behind the greedy behaviour of the heuristic. Therefore, the output of the randomised algorithm is unlikely to provide a good solution. Also, note that this procedure could be run multiple times, but it is likely that all the solutions generated could be significantly worse than the one provided by the original heuristic. Not to lose the benefits of greedy behaviour of the heuristic, we applied biased randomisation in construction step to improve the solution of IG algorithm.

The randomisation process can be classified by using two types of probability distribution, which are uniform randomised distribution (symmetric) and a biased randomised distribution (non-symmetric). However, most researchers studied biased randomised distribution.

Figure 3.7 shows a comparison between Uniform Randomised and Biased Randomised. The Uniform Randomised assigns equal chance for all items in the lists to be selected randomly, while Biased-Randomised needs to assign a probability by using some parameters and rearrange the items list based on those probabilities. The highest probability will be the first candidate and the lowest probability will be the last candidate in the list. Then, select the item that is the first candidate in the list.

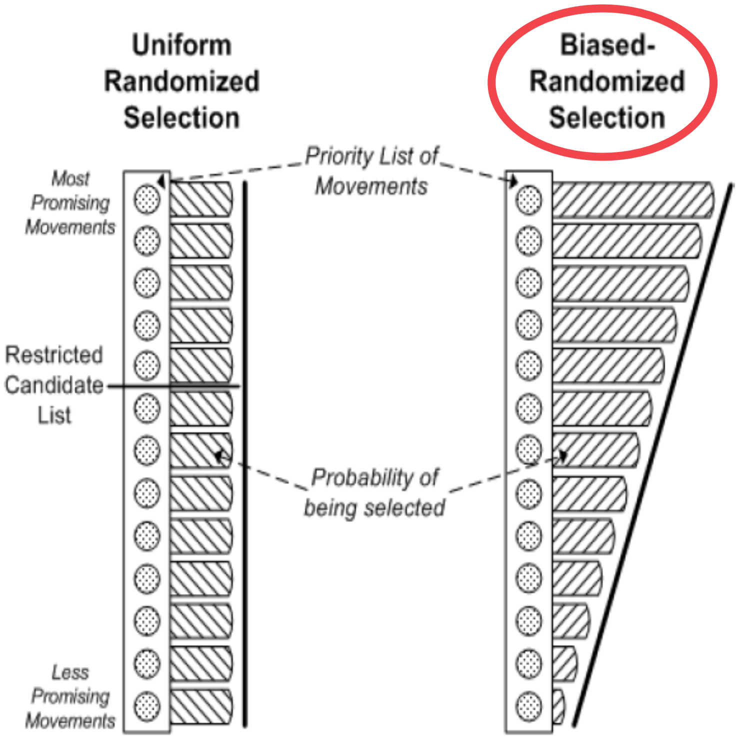

Figure 3.7 : Uniform randomisation vs. biased randomisation (Cáceres-Cruz, 2013)

Figure 3.8 shows where we add the biased randomisation concept into the IG algorithm which shown in big **red** arrow. The IG-BR for RSP process starts working on the destruction phase, then we add the biased randomised algorithm at the end of destruction phase for reordering the construction set. After that, we work on the construction phase by reinserting the first element in a construction set back into the remaining set on every position to find the best position that provides the best solution. All of these processes are repeated until the stopping criteria are met.

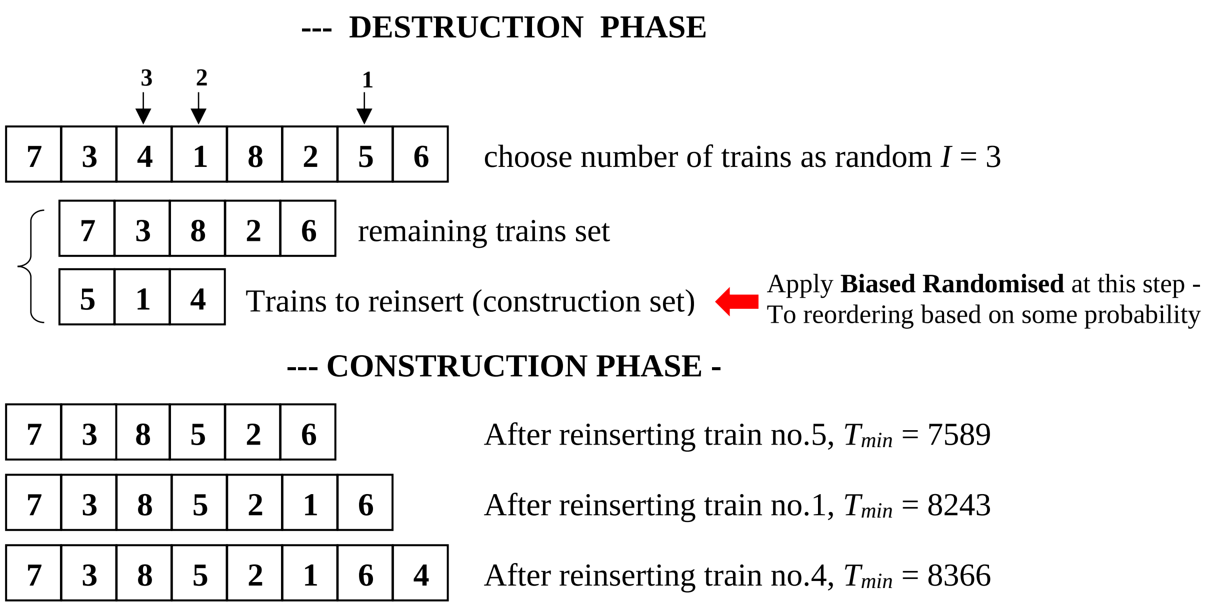

Figure 3.8 : Apply IG biased randomised (Ruiz et al., 2007)

We proposed an extension by applying biased randomised technique into the IG algorithm in order to improve the solution of the deterministic RSP under the uncertainties delay. The main procedure is described as follows: (a) Initial solution based on given timetables in our case studies which comprises of real-world data from the UK and Thailand Rail network, (b) IG is used to solve the RSP by using two main phases (destruction and construction), (c) the use of biased randomised algorithm to improve the resulting solution (Algorithm 3.2).

Algorithm 3.2 : Procedure for Deterministic RSP IG-BR Algorithm

---

1: **Procedure for Deterministic RSP IG-BR Algorithm** (r', r^D, r^R, d, b, β)
▻ r': Initial solution
▻ r^D: Partial sequence to reconstruct
▻ r^R: Trains to reinsert
▻ d: Random chosen number of trains
▻ b: Number of blocks
▻ β: Parameter for biased randomised - range of parameter β is [0,1]
2: r': *GenerateInitialSolution*(); ▻ *Initial solution*
3: **while** execution_time < 400 **do** ▻ *Execution time 400 seconds*
4: **for** i **=** 1 to b **do** ▻ *Run for all block*
5: set r^R = empty
6: **for** i **=** 1 to d **do** ▻ *Destruction step*
7: r^R <- remove one node randomly from r' and insert it in r^R;
8: $r^D = r'$ <- the remaining set of trains
9: **end for**
10: re-ordering r^R by using β to assign a probability ▻ *Biased randomised*
11: **for** j = 1 to d **do** ▻ *Construction step*
12: r^{ans} = best solution obtained after insert train from r^R in all possible position of r^D
13: **if** r^{ans} met with all railway constraints **then** ▻ *check all constraints*
14: solution (r^{ans}) = calculate minimise a total delay (r^{ans})
15: **end if**
16: **end for**
17: **end for**
18: **if** solution (r^{ans}) < solution (r') **then** ▻ *Minimise the total delay*
19: solution (r^{best}) = solution (r^{best});
20: **else if**
21: solution (r^{best}) = solution (r');
22: **end if**
23: **end while**
24: **end procedure**

Algorithm 3.2 shows pseudocode for deterministic RSP IG-BR Algorithm. Firstly, the process starts by creating an initial solution from the rail timetable (Pseudocode: line 2). We consider the deterministic disruption situation, so the amount of the delay is known beforehand, and we add it together with the initial input.

Then, we start to implement the destruction phase of IG to find a new feasible solution (Pseudocode: line 6 to line 9). However, at the end of the destruction phase, we apply the biased randomised algorithm to reorder the construction set by using the geometric probability distribution, which can give a probability to each train in the construction set was calculated by the following equation ($equation$ 3.13).

$$\text{Probability of train } i \text{ in block } m = 1 - \frac{\text{Running time } (p_{im})}{\text{Total running time } (\sum p_{im})} \qquad (3.13)$$

Then, rearrange the trains based on those probabilities from high to low, so a smaller running time is more likely to be selected than longer running times before continuing to the next step (Pseudocode: line 10).

Then, at the construction phase, we reinsert the construction set back into the remain set following the basic IG concept, which inputs in all possible positions, and calculates to find the best sequence that provides the best solution (Pseudocode: line 11 to line 16). In addition, we also need to check that operational constraints such as departure time, running/dwell time and headway constraints are still valid (Pseudocode: line 13 to line 15)

Following the destruction phase, construction phase and the biased randomised phase, we use simplest acceptance criteria to check that a new timetable has provided a better solution or not, as a smallest total delay and set this as the current solution for the next iteration (Pseudocode: line 18 to line 22). After that, the new schedule will be sent to the rail control centre to manage the rail process. Finally, we have repeated all process until the stopping criteria of 400 seconds are met.

3.5. Computational experiments

In this section, we evaluate the performance of the optimisation model and IG-BR, which is the proposed solution method in this chapter, by comparing the results of IG-BR with MILP obtained by CPLEX optimisation software package and/or historical data from the company. We selected the CPLEX as one of the solution methods to solve the deterministic RSP because our case studies have not been used before in the literature. Due to lack of the results for evaluate the performance of our proposed solution methods, it is reasonable to use the CPLEX software package which can provide the optimal solution for discuss on the performance analysis. We experimented on the real-world case studies from two countries as the UK and Thailand Rail networks and considered the constraints as follows:

(i) Southeastern train operating company, UK that consisted of 27 trains and 135 blocks. This part of the network has a complex infrastructure with bottleneck areas and dense traffic on interconnected lines for passengers in and out of London. Moreover, the UK rail network uses four-aspect signalling for rail dispatching management. In addition, the minimum headway between two consecutive trains set as 150 seconds due to the UK railway regulation.

(ii) State Railway of Thailand that consisted of 101 trains and 332 blocks. The data were collected from the whole of the Thailand rail network, including 4 main lines which are Northern line, Eastern line, North-eastern line, and Southern line. All trains have start and end at Bangkok stations, which provide the dense of traffic for passengers in and out of Bangkok. Moreover, Thailand rail network uses two-aspect signalling for rail dispatching management. In addition, there is no minimum headway between two consecutive trains because they only operated based on the signalling regulation.

In the case of deterministic delay on RSP, we know the exact running time or dwell time delay on each block and late departure at the origin block before the simulation starts.

We implemented IG-BR heuristic in C# programming language, then compared the result with CPLEX commercial optimisation tool. We executed all experiments on the same personal computer with Intel core i7-4500U CPU, 1.80GHz-2.4GHz, and 8GB RAM. Moreover, all computational experiments were set for stopping condition as 400 seconds because Southeastern Service Disruption Procedure (SSDP) suggests a time frame of 15 minutes to update the service or schedule.

3.5.1. Southeastern train company, United Kingdom

The set of data was collected from Southeastern train company covering London Bridge area, Kent and South East UK (Khosravi, 2013). We selected this case study because of the complexity of infrastructure, bottleneck area, and the dense traffic on interconnected lines for passengers in and out of London, East Sussex and the Channel Tunnel as shown in Figure 3.9. The train service information was collected from public passenger timetables; however, details of the exact routes were not available. Therefore, we used the National Rail Enquiries (2013) website to define platforms and routes information for our schedule and reschedule.

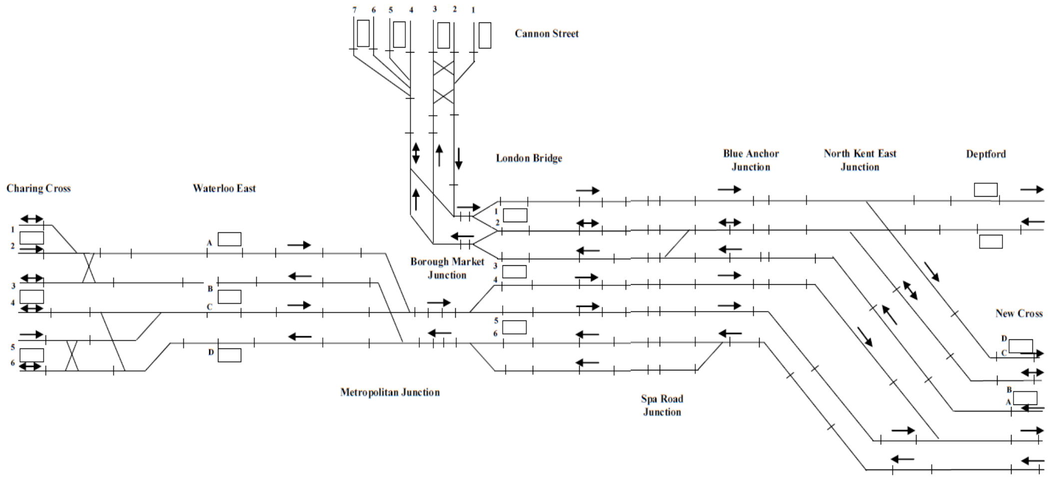

Figure 3.9 : London Bridge diagram developed for scheduling problem (Khosravi, 2013)

This area of network is about 15 km long with a total of 28 platforms which includes many busy stations such as London Charing Cross, London Waterloo, London Cannon Street, New Cross and Deptford. The most important characteristic is the signalling system. The UK rail network uses four-aspect signalling for rail dispatching management.

For the UK test instances, the case study focused on the off-peak services with the timetable cycle every 30 minutes, 135 blocks and 27 trains in the network. Southeastern operating company has a Southeastern Service Disruption Procedure (SSDP) to handle disruptions and it classifies disruptions into three types which are minor disruption (delay less than 15 minutes), general disruption (delay between 15 to 30 minutes) and major disruption (delay more than 30 minutes), however there is no exact number for disrupted trains in this classification. According to this case study, there are 27 trains in a cycle and Khosravi (2013) has classified the number of trains into three types which are minor disruption, disturbing 1 to 5 trains, general disruption, disturbing 6 to 12 trains and major disruption disturbing 13 to 27 trains. According to the railway safety regulation, headway time between two consecutive trains is 150 seconds and the rail network has control by four-aspect signalling.

3.5.1.1. Test instances

In the experiment on the deterministic RSP, we used 4 sets of instances (58 test instances in total) from the UK case study provided by Khosravi (2013); each set of data was considered on different types and sizes of disruption. For the disruption types, we considered late departure from the origin, longer running time and longer dwell time at the stations. Moreover, on the type of disruption, we combined minor and general disruptions together for the first 2 sets of instances (Tables 3.1-3.2) and major disruptions for the other 2 sets of instances (Tables 3.3-3.4). In addition, block delay, which refers to longer running time/dwell time and departure delay, which refers to late departure time are provided in minutes. Tables 3.4 to 3.7 below show details of the set of instances.

Table 3.4 shows the first set of experiments which consider longer running time and longer dwell time added into the running time. The set of data includes 6 minor and 6 general disruption instances. Minor disruption delays are chosen in the set {5, 10, 15} minutes and general disruption delays are chosen in the set {20, 25, 30} minutes. Furthermore, the extra time is added into 1 or 2 blocks which then affect 3 or 5 trains for minor disruption and 9 or 12 trains for general disruption.

Disruption Type	# of Instance(s)	Block Delay	# of Block(s)	# of Train(s)
Minor	6	5, 10, 15	1, 2	3, 5
General	6	20, 25, 30	1, 2	9, 12

Table 3.4 : Longer running/dwell times - minor and general disruption - Deterministic (UK)

Table 3.5 shows the second set of experiments which considers late departure time, added into the departure time. The set of data includes 9 minor and 6 general disruption instances. Minor disruption delays are chosen in the set {5, 10, 15} minutes and general disruption delays are chosen in the set {20, 25} minutes. Furthermore, the extra time is added affects 1, 3 or 5 trains on minor disruption and 6, 9 or 12 trains on general disruption. However, departure delays should be less than 25 minutes because departure delays larger than 25 minutes result in shifting trains out of the original cycle.

Disruption Type	# of Instance(s)	Departure Delay	# of Train(s)
Minor	9	5, 10, 15	1, 3, 5
General	6	20, 25	6, 9, 12

Table 3.5 : Late departures - minor and general disruption - Deterministic (UK)

Table 3.6 shows the third set of experiments which considers longer running time and longer dwell time added into the running time. The set of data comprises 15 major disruption instances which are chosen in the set {40, 50, 60, 70, 80} minutes. Furthermore, we add extra time into {4, 5, 6} blocks which affects all 27 trains in a cycle.

Disruption Type	# of Instance(s)	Block Delay	# of Block(s)	# of Train(s)
Major	15	40, 50, 60,70, 80	4, 5, 6	27

Table 3.6 : Longer running/dwell times - major disruption – Deterministic (UK)

Table 3.7 shows the fourth set of experiments which considers 16 major disruption instances with both longer running/ dwell times and late departures. We created block delays for 40 or 50 minutes occurring on 4 or 6 blocks which affected the majority of services. However, departure delays were less than 25 minutes because departure delays larger than 25 minutes resulted in shifting trains out of the original cycle. Therefore, we avoided generating any departure delays longer than 20 minutes. Departure delays as 10 and 20 minutes were tried for different numbers of trains in set {3, 5, 9, 12}.

Disruption Type	# of Instance(s)	Block Delay	# of Block(s)	Departure Delay	# of Train(s)
Major	8	40	4, 6	10, 20	3, 5, 9, 12
	8	50	4, 6	10, 20	3, 5, 9, 12

Table 3.7 : Longer running/dwell times and Late departures - major disruption - Deterministic (UK)

3.5.1.2. Performance analysis

The computational experiments were conducted to compare CPLEX, and IG-BR results based on 400 seconds within the time frame of 15 minutes determined in SSDP for the update the service. We selected CPLEX as one of our solution methods because it is commercial software which runs based on an exact method. Normally, the CPLEX is used to generate the optimal solution; however, it also needs large computation time to reach the optimal solution. In the UK case study, we computed the CPLEX to find the optimal solution, but the program required more than 7 hours of execution time. According to SSDP, we set the model to reschedule and provide the new schedule within 400 seconds after the delay occurred, so we do not have enough time and the CPLEX must stop before reach the optimal point. Therefore, comparison between CPLEX and IG-BR will help to provide a clear view when we evaluate the efficiency of the IG-BR.

Tables 3.8-3.11 show the experimental results using CPLEX and IG-BR to solve the UK case study. In each table, all columns before the CPLEX column show the detail of each instance and all delays are shown in **minutes**. For the CPLEX column, we only provide a total train delay. However, for columns IG-BR, we present the results of total delay and the percentages of improvement between IG-BR and CPLEX by using relative deviation (sub-column RD (%) compared with CPLEX) for all instances. Finally, at the bottom row of each table, we calculated an average of the total delay and RD which can provide a clearer understanding of the results. The average total delay is calculated by $\sum_{i=1}^{n} T_i/n$, where T_i is the total delay of each instance, i indicates the number of instances and n is the total number of instances.

Moreover, the relative deviation (RD%) which is the improvement gap used to present the efficiency of our proposed methodology was calculated by the following equation:

$$\text{RD\%} = \frac{\text{baseline solution} - \text{algorithm solution}}{\text{baseline solution}} \; x\, 100 \quad (3.13)$$

where the baseline solution is considered as the solution method that we need to "compared with", and algorithm solution is considered as the current solution method that we need to measure the performance. Therefore, the positive number of RD (%) means that the result of our proposed method can provide a better solution than the original results, while negative results mean there are no improvements.

Block Delay	# of Block(s)	# of Train(s)	Instance	CPLEX	IG-BR	RD (%) of IG-BR compared with CPLEX
				Delay	Delay	
5	1	3	P01	107.83	**97.67**	**9.42**
	2	5	P02	143.67	**140.33**	**2.32**
10	1	3	P03	167.00	168.67	-1.00
	2	5	P04	**281.83**	314.83	-11.71
15	1	3	P05	260.17	**247.33**	**4.93**
	2	5	P06	314.67	**313.33**	**0.42**
20	1	9	P07	**1028.00**	1099.83	-6.99
	2	12	P08	**661.75**	705.50	-6.61
25	1	9	P09	**1272.67**	1471.17	-15.60
	2	12	P10	**1823.00**	1829.00	-0.33
30	1	9	P11	**1021.00**	1023.33	-0.23
	2	12	P12	**1769.58**	2052.00	-15.96
Avg Delay				**737.60**	788.58	**-6.91**

Table 3.8 : Deterministic Longer running/dwell times - minor and general disruption: IG-BR Results

Table 3.8 shows the IG-BR results of deterministic RSP considered for minor and general delays on longer running time and dwell time. The best solution for each instance is indicated in **bold**. However, in the overall results, the comparison between CPLEX and IG-

BR shows that using CPLEX to solve the RSP decreased the average total train delay by 6.91% more than the IG-BR. It should be noted that CPLEX was computed only 400 seconds, therefore CPLEX cannot achieve the optimal solution. Therefore, the IG-BR provide lower average delay in some test instances, which means IG-BR also can be used for solving RSP in small cases but at slightly lower efficiency than CPLEX.

Late Departure	# of Train(s)	Instance	CPLEX Delay	IG-BR Delay	RD (%) of IG-BR compared with CPLEX
5	1	R01	**61.83**	70.83	-14.55
	3	R02	**59.83**	65.50	-9.47
	5	R03	**73.17**	75.17	-2.74
10	1	R04	**70.67**	78.50	-11.09
	3	R05	**82.17**	83.50	-1.62
	5	R06	**138.00**	143.83	-4.22
15	1	R07	**67.66**	68.67	-1.49
	3	R08	**119.33**	127.33	-6.70
	5	R09	**167.83**	171.33	-2.08
20	6	R10	**182.33**	184.00	-0.92
	9	R11	**240.50**	250.67	-4.23
	12	R12	**334.17**	347.67	-4.04
25	6	R13	**230.67**	240.33	-4.19
	9	R14	**275.67**	281.67	-2.18
	12	R15	**376.67**	385.17	-2.26
Avg Delay			**165.37**	171.61	**-3.78**

Table 3.9 : Deterministic Late departures - minor and general disruption: IG-BR Results

Table 3.9 shows the IG-BR results of deterministic RSP considered on minor and general delays of late departure times. The best solution for each instance is indicated in **bold**. As expected, the CPLEX outperforms among the other algorithms in terms of average total train delays by 3.78% more than IG-BR. The CPLEX was implemented based on exact methods which provided a good solution for small sizes of disruptions. Therefore, the CPLEX was the best choice for rescheduling with the aim to minimise the total train delay for only minor and general disruptions. Moreover, the result of CPLEX was not optimal because the program was stopped at 400 seconds. We computed the solution based on this time frame to evaluate the performance of IG-BR and found only a small gap between IG-BR and CPLEX.

Block Delay	# of Block(s)	Instance	CPLEX	IG-BR	RD (%) of IG-BR compared with CPLEX
			Delay	Delay	
40	4	HP01	7706.00	**7379.33**	**4.24**
	5	HP02	7432.00	**7260.75**	**2.30**
	6	HP03	7619.50	**7250.47**	**4.84**
50	4	HP04	9396.67	**8836.87**	**5.96**
	5	HP05	9342.33	**9150.20**	**2.06**
	6	HP06	8529.00	**7548.50**	**11.50**
60	4	HP07	11574.83	**10429.50**	**9.90**
	5	HP08	9309.58	**8795.90**	**5.52**
	6	HP09	9570.67	**8497.00**	**11.22**
70	4	HP10	13182.67	**12187.30**	**7.55**
	5	HP11	10716.58	**9656.60**	**9.89**
	6	HP12	13006.42	**11795.70**	**9.31**
80	4	HP13	15491.00	**14262.90**	**7.93**
	5	HP14	14858.67	**13645.60**	**8.16**
	6	HP15	15646.83	**14103.50**	**9.86**
Avg Delay			10892.18	**10053.34**	**7.70**

Table 3.10 : Deterministic Longer running/dwell times - major disruption: IG-BR Results

Table 3.10 presents the IG-BR results of deterministic RSP considered for major delays of longer running time and dwell times. The best solution for each instance is indicated in **bold**. The results of this set of instances showed that IG-BR reduced the average of total train delays by 7.70% more than CPLEX. Noted that CPLEX was executed only 400 seconds and not reach the optimal solution; therefore, the IG-BR performance better for solving the RSP when considering major disruptions of running time and dwell time delay within the time limit.

Block Delay	# of Block(s)	Departure Delay	# of Train(s)	Instance	CPLEX	IG-BR	RD (%) of IG-BR compared with CPLEX
					Delay	Delay	
40	4	10	3	PR01	7524.00	**7379.92**	**1.91**
			5	PR02	7455.67	**7285.42**	**2.28**
		20	9	PR03	7599.17	**6975.83**	**8.20**
			12	PR04	7823.50	**7107.00**	**9.16**
40	6	10	3	PR05	**6830.00**	6930.50	-1.47
			5	PR06	7773.56	**7002.83**	**9.91**
		20	9	PR07	6518.50	**6465.85**	**0.81**
			12	PR08	8088.50	**7458.50**	**7.79**
50	4	10	3	PR09	9462.17	**9346.13**	**1.23**
			5	PR10	9615.75	**9097.40**	**5.39**
		20	9	PR11	9589.00	**7672.54**	**19.99**
			12	PR12	9863.33	**9194.50**	**6.78**
50	6	10	3	PR13	9346.42	**8400.54**	**10.12**
			5	PR14	**9737.17**	9741.20	-0.04
		20	9	PR15	8609.00	**8529.00**	**0.93**
			12	PR16	8286.92	**8012.83**	**3.31**
Avg Delay					8382.67	**7912.50**	**5.61**

Table 3.11 : Deterministic Longer running/dwell times and Late departures - major disruption: IG-BR Results

Table 3.11 indicates that the IG-BR results of deterministic RSP considered on major disruption of late departure times. The best solution for each instance is indicated in **bold**. The overall average of the total train delay of IG-BR was lower than CPLEX by 5.61%. According to 400 seconds computation time limit, we can claim that IG-BR was more effective than CPLEX for solving the set of instances considering major delay for all types of disruption which are travel time delay, dwell time delay and late departure time delay because CPLEX was terminated by the time limit before achieving the optimal value.

The SSDP separated the disruptions into 3 types depending on the size of disruption as follows:

- Minor disruptions - the delay less than 15 minutes and affected 1 to 5 trains.
- General disruptions - the delay between 15-30 minutes and affected 6 to 12 trains.
- Major disruptions - the delay more than 30 minutes and affected 13 to 27 trains.

Therefore, we summarised the results based on the disruption types as follows:

Disruption Type		CPLEX Delay	IG-BR Delay	RD (%) of IG-BR compare with CPLEX
Minor and General	Longer Running/dwell times	**737.60**	788.58	-6.91
	Late Departures	**165.37**	171.61	-3.78
Average Delay		**451.48**	480.10	**-6.34**

Table 3.12 : Summary of Experimental result of IG-BR - Deterministic (UK) - Small

Table 3.12 shows a summary of results from Tables 3.8 and 3.9 which considered minor and general delay that can be group together as the small size disruption. When we compared between CPLEX and IG-BR, the best solution was provided by CPLEX which decreased the average total train delays by 6.34%. According to the basic knowledge from the literature review, the CPLEX implement based on the exact method had high performance to solve small sizes of problem. Please note that all experiments were based on 400 seconds of computation time and the solution of CPLEX was optimal.

Disruption Type		CPLEX Delay	IG-BR Delay	RD (%) of IG-BR compare with CPLEX
Major	Longer Running/dwell times	9637.42	**8982.92**	6.79
	Late Departures & Longer Running/dwell times	5044.45	**4731.51**	6.20
Average Delay		7340.94	**6857.21**	**6.59**

Table 3.13 : Summary of Experimental result of IG-BR - Deterministic (UK) - Large

Table 3.13 shows the summary of results from Tables 3.10 and 3.11 which considered on the large disruption instances. It shows that CPLEX provided a bad solution for solving the major disruption for both late departures delay and longer running time/dwell time delay due to the complexity of the problem. The program needed more than 400 seconds to achieve a better solution. The results showed that IG-BR reduced average total train delays by 6.59% when compared to the CPLEX in the case of large sized disruptions within the limit of the computation time.

Disruption Type		CPLEX Delay	IG-BR Delay	RD (%) of IG-BR compare with CPLEX
Minor and General	Longer Running/dwell times	**737.60**	788.58	-6.91
	Late Departures	**165.37**	171.61	-3.78
Major	Longer Running/dwell times	10214.26	**10053.34**	1.58
	Late Departures & Longer Running/dwell times	7892.23	**7912.50**	-0.26
Average Delay		5044.45	**4731.51**	**6.20**

Table 3.14 : Summary of Experimental result of IG-BR - Deterministic (UK) - All

Table 3.14 presents the overall picture of all results in section 3.5.1.2. CPLEX was effective for solving minor and general disruption RSPs while IG-BR was a better choice to solve major disruption RSPs. On the average of the total train delays, IG-BR improved the quality of solution by 6.20%. The 400 seconds time frame required by SSDP to update the schedule was not long enough for CPLEX to achieve the optimal value. The IG-BR was more effective to use for the deterministic RSP.

In addition, in real-life operation of the railway network, the we need to deal with mixed types of minor, general and major disruption together, so we highly recommend the IG-BR that provided the best solution in terms of overall average total train delays.

3.5.2. State Railway of Thailand

The State Railway of Thailand is the state-owned rail operator under the jurisdiction of the Ministry of Transport in Thailand. The network operates in 47 provinces and served 35 million passengers annually in 2018. Data were collected from the whole of the Thailand rail network, including 4 main lines which are Northern line (from Bangkok to Chiang Mai), Eastern line (from Bangkok to Aranyaprathet or Pattaya), North-eastern line (from Bangkok to Ubon Ratchathani or Nong Khai), and Southern line (from Bangkok to Butterworth or Kanchanaburi). In addition, all train lines start and end at the same station in Bangkok, which means that the station area in Bangkok is very busy. If disruption occurs, it will quickly propagate into the whole network due to train interconnectivity. All trains are operated following the two-aspect signalling system which means the headway between two consecutive trains equal to the running time of the previous train.

A map of the Thailand railway network is shown in Figure 3.10. Information regarding train service, route and timetable was extracted from the Thai Rail service website which is open to the public (State Railway of Thailand, 2019). However, some data such as tracks, regulation, number of platforms, running time and travelling distance are not shown on the website. This information was directly collected from the State Railway of Thailand (2019).

Figure 3.10 : Thailand Rail Map (Thailand Trains, 2019)

For the Thailand test instances, the case study focused on the one-day timetable, with trains starting from 5.00 am until 11.59 pm. The whole network includes 332 blocks/stations and 101 trains (51 outbound trains, and 50 inbound trains). In addition, the Thailand rail network has considered blocks as the distance between one station and the next. Moreover, State Railway of Thailand (2019) has classified disruptions into three types as minor disruption (delay less than 30 minutes), general disruption (delay between 30 and 60 minutes) and major disruption (delay more than 60 minutes).

3.5.2.1. Test instances

In the experiments on the deterministic RSP, the State Railway of Thailand (2019) provided one-month historical data between 01/02/2019 and 28/02/2019, which means we have 28 test instances in total as shown in Table 3.15. Furthermore, this table also provides historical data about total delay for the whole Thailand rail network, but there is only information about the length of the delays and the corresponding stations where delays occurred. Therefore, we cannot classify cause of disruptions, and also cannot classify all delays separately out as minor, general, or major delay. In each test instance, we have considered on the mix types of disruption which are minor, general, and major disruptions together as they provided in the historical data. In addition, Thailand's rail operates by using a manual system which follows the first come first serve (FCFS) rule, and human decision making. Therefore, the historical data are based on the FCFS rule implemented by dispatchers. Hence, efficiency of the system can be improved significantly. Test instances are considered for 101 trains and 332 blocks. Thus, the size of the problem is very large and details of test instances cannot be provided due to confidentiality issues.

Date	Instance	**Historical Data**
		Delay
1/2/2019	THAF101	2477
2/2/2019	THAF102	2138
3/2/2019	THAF103	1830
4/2/2019	THAF104	1681
5/2/2019	THAF105	2549
6/2/2019	THAF106	1750
7/2/2019	THAF107	2329
8/2/2019	THAF108	3186
9/2/2019	THAF109	2143
10/2/2019	THAF110	2180
11/2/2019	THAF111	2712
12/2/2019	THAF112	2472
13/2/2019	THAF113	2440
14/2/2019	THAF114	1807
15/2/2019	THAF115	2903
16/2/2019	THAF116	3582
17/2/2019	THAF117	3516
18/2/2019	THAF118	2895
19/2/2019	THAF119	2880
20/2/2019	THAF120	2791
21/2/2019	THAF121	3798
22/2/2019	THAF122	3658
23/2/2019	THAF123	3772
24/2/2019	THAF124	4597
25/2/2019	THAF125	3974
26/2/2019	THAF126	4873
27/2/2019	THAF127	3385
28/2/2019	THAF128	3023
Average Delay		2905.04

Table 3.15 : Thailand test instances and historical data - Deterministic (Thai)

3.6.2.2. Performance analysis

The computational experiments compare the historical data and the result of MILP and IG-BR obtained within the 400 second time limit adapted from the UK rail network time frame for updating schedules. We selected CPLEX as one of our solution methods because it is a commercial software package which runs based on exact methods and helps to provide a clear view when we compare the efficiency of the solution methods. For large size test instances, CPLEX used more than 15 hours to find the optimal solution, so the program was stopped at 400 seconds to comply with the real-world situation. Therefore, the comparison between historical data, CPLEX and IG-BR showed the efficiency of the IG-BR.

In the historical data, each test instance consists of three types of disruption which are minor, general, and major disruptions mixed together. Therefore, the mix types of delay have been added into the original timetable based on the delay information which provided from State Railway of Thailand.

Table 3.16 shows the experimental result for using historical data, CPLEX and IG-BR. In this table, the first two columns before the historical data column show the detail of each instance and all delays are provided in **minutes**. In the historical data column, we provided a total delay of all trains collected from the Thailand rail company. For the CPLEX and IG-BR columns, we provided a total delay and RD (%) to compare the proposed methods with the historical data. Then, we also provided RD (%) to evaluate the performance between CPLEX and IG-BR. Finally, in the bottom row of the table, we calculated an average of the total delay and RD. In addition, RD (%) was used to show improvement of the solution method.

Date	Instance	Historical Data	CPLEX		IG-BR		RD (%) compared with CPLEX
		Delay	Delay	RD (%) compared with historical data	Delay	RD (%) compared with historical data	
1/2/2019	THAF101	2477	2121	14.37	**1973**	**20.35**	6.98
2/2/2019	THAF102	2138	2035	4.82	**1998**	**6.55**	1.82
3/2/2019	THAF103	1830	1803	1.48	**1789**	**2.24**	0.78
4/2/2019	THAF104	1681	1325	21.18	**1154**	**31.35**	12.91
5/2/2019	THAF105	2549	2379	6.67	**2247**	**11.85**	5.55
6/2/2019	THAF106	1750	1258	28.11	**1234**	**29.49**	1.91
7/2/2019	THAF107	2329	2188	6.05	**2107**	**9.53**	3.70
8/2/2019	THAF108	3186	2648	16.89	**2422**	**23.98**	8.53
9/2/2019	THAF109	2143	1907	11.01	**1895**	**11.57**	0.63
10/2/2019	THAF110	2180	1574	27.80	**1535**	**29.59**	2.48
11/2/2019	THAF111	2712	1982	26.92	**1825**	**32.71**	7.92
12/2/2019	THAF112	2472	1830	25.97	**1794**	**27.43**	1.97
13/2/2019	THAF113	2440	1874	23.20	**1713**	**29.80**	8.59
14/2/2019	THAF114	1807	1807	0.00	**1780**	**1.49**	1.49
15/2/2019	THAF115	2903	2458	15.33	**2265**	**21.98**	7.85
16/2/2019	THAF116	3582	3030	15.41	**2877**	**19.68**	5.05
17/2/2019	THAF117	3516	2740	22.07	**2554**	**27.36**	6.79
18/2/2019	THAF118	2895	2645	8.64	**2321**	**19.83**	12.25
19/2/2019	THAF119	1880	1457	22.50	**1398**	**25.64**	4.05
20/2/2019	THAF120	2791	1861	33.32	**1801**	**35.47**	3.22
21/2/2019	THAF121	3798	3050	19.69	**2821**	**25.72**	7.51
22/2/2019	THAF122	3658	3421	6.48	**3293**	**9.98**	3.74
23/2/2019	THAF123	3772	3543	6.07	**3355**	**11.06**	5.31
24/2/2019	THAF124	3597	3260	9.37	**3180**	**11.59**	2.45
25/2/2019	THAF125	3974	3477	12.51	**3321**	**16.43**	4.49
26/2/2019	THAF126	4873	4495	7.76	**4288**	**12.00**	4.61
27/2/2019	THAF127	3385	3385	0.00	**3275**	**3.25**	3.25
28/2/2019	THAF128	3023	2584	14.52	**2447**	**19.05**	5.30
Average Delay		2833.61	2433.46	14.12	**2309.36**	**18.50**	**5.10**

Table 3.16 : Deterministic Thai - IG-BR Results

Table 3.16 presents the IG-BR results of deterministic RSP for the Thailand case study. The best solution for each instance is indicated in **bold**. Due to the size of Thailand rail network, the CPLEX would need a long computation time to find the optimal solution, so we set the time limit as 400 seconds to serve the real-world RSP situation. The results showed that when we applied our proposed optimisation model and solution methods as CPLEX and IG-BR to the real-world data from Thailand, an improvement was shown on the rail management process. The experimental results in all instances show that both CPLEX and IG-BR improved the current solution used by the Thai railway company. CPLEX reduced the average total train delays by 14.12% and IG-BR reduced the average total train delays by 18.50%. Moreover, comparison between IG-BR and CPLEX showed that average total train delays of BR-IG were lower than CPLEX at 5.10%.

3.6. Chapter Conclusion

This chapter developed IG-BR algorithm to improve the solutions in deterministic RSP under three disruption types which are late departure, longer running time, and longer dwell time by minimising total train delays which is a major criterion directly related to the efficiency of train traffic management. According to size and complexity of the RSP, the problem is very hard to solve by using the exact method, because it required very huge computation time to solve a problem. Then, many methodologies based on heuristics and metaheuristics were developed to deal with large-scale COPs to support the decision-making management. Therefore, we have selected the Iterated Greedy (IG) algorithm which is one of the well-known heuristics that very successfully and effectively to solve many OR problems to solve the RSP. Moreover, the basic IG algorithms showed greater flexibility to implement with COPs, because of the parameter free process which make it easy to combine with state-of-the-art techniques. Then, the biased randomisation technique was selected to combine with IG algorithm because it gave good results in the other scheduling problems and improved the quality of heuristics.

For the deterministic optimisation model, we developed the optimisation model following the formulation of Khosravi (2013) based on a blocking job shop scheduling problem. The objective of the optimisation model is to minimise the total weighted tardiness which is equivalent to total train delays.

We proposed the solution methods called IG-BR. The proposed solution method of IG-BR was performed by following two main phases of IG heuristic algorithm which are destruction phase and construction phase. At the end of the destruction phase, the biased randomisation technique with geometric probability distribution was provided to reorder the constructed set before reinserting back in the construction phase. This added more diversification to the basic IG algorithm.

We used the proposed optimisation model and algorithm to solve the real-world case studies from Southeastern train operating company, UK (Khosravi, 2013) and State Railway of Thailand. There were 400 seconds of computation time limit to update the schedule as suggested by SSDP. In the UK test instances, the overall results showed that IG-BR was the best solution for the deterministic RSP; however, for the minor and general disruption set of instances, the solution from CPLEX was better than IG-BR. For the Thailand case study, the experimental results showed that the IG-BR provided a better solution compared to the company’s historical solution. Therefore, the IG-BR algorithm is more effectively for solving RSP in large and complex problems.

However, this proposed IG-BR can be improved by combining with other state-of-the-art techniques. The LS process introduces local changes into the heuristic, which decreases the computation time at the construction phase and improve the search mechanism in the solution generation process. For this reason, in the next chapter, we developed a LS heuristic to the IG-BR algorithm.

Chapter 4: Biased Randomised Iterated Greedy with Local Search (BR-IG-LS) for deterministic RSP

4.1. Introduction

In the previous chapter, the result of IG-BR when compared with CPLEX showed small improvement gap on major disruption instances, therefore this chapter will propose the other solution methods which can improve the results of IG-BR. Thus, we selected the most popular local search heuristics which can help to increase the search space size and improve the solutions of IG algorithm in other COPs.

In this chapter, we propose the BR-IG-LS to solve RSPs that deal with a deterministic delay and improve the results of IG-BR in Chapter 3. This chapter discusses the basic concept of Local Search (LS) as a heuristic method. LS usually combines with the IG algorithm to improve efficiency in many optimisation problems such as Freight train scheduling (Yuan et al., 2008), and Flow shop scheduling (Ruiz et al., 2008). Ruiz et al. (2007) were the first to use the optional LS process to improve the IG algorithm on the permutation flow shop scheduling problem and provided better results when compared to basic IG methods. After that, the IG-LS has been used with success to solve many COPs; however, the problem which is the most similar to RSP is a freight train scheduling problem. Yuan et al. (2008) were the first to adapt IG-LS to solve freight train scheduling but they only considered freight RSP, not passenger RSP. The freight RSP was addressed by Fügenschuh et al. (2008), who used commercial integer linear programming to solve a small test case from Deutsche Bahn AG. They also applied IG heuristic with extension of LS to solve the problem and the results showed that IG heuristic with extension of LS was more powerful for the real-life freight RSP and suitable to use in large instances. Moreover, Kang et al. (2013) claimed that LS improved speed and performance of the IG algorithm by increasing the solution search space. Therefore, in this chapter, we propose an BR-IG-LS algorithm to solve RSP that deals with a deterministic delay.

This chapter first discusses the concept of BR-IG-LS and then shows how to use BR-IG-LS to solve a deterministic RSP. The experimental results of the UK and Thailand case studies show the efficiency of the proposed methods when compared with the results from the previous chapter and improvement when a LS heuristic is added into the IG-BR.

The chapter is structured as follows: section 4.2 presents this chapter's contributions. The proposed BR-IG-LS algorithm to solve RSP with deterministic disruption delays is presented in section 4.3, while all the experimental results are shown in section 4.4 which separates two sets of case studies (UK & Thailand). Section 4.5 is the conclusion of this chapter.

4.2. Contribution

Regarding the literature, most researchers who studied the IG algorithm suggested that the local search heuristic improved the efficiency of the basic IG algorithm. The IG-LS has been used successfully in similar optimisation problems such as Task assignment (Harish et al., 2014), Task allocation (Kang et al., 2013), Parallel machine scheduling (Ying et al., 2010), Freight train scheduling (Yuan et al., 2008), and Flow shop scheduling (Ruiz et al., 2008). As IG-LS has been adapted successfully on many COPs, it is reasonable to apply LS to improve the solution of the IG-BR algorithm, therefore the LS was chosen to combine with IG-BR algorithm to improve the quality of solution. Moreover, Kang et al. (2013) claimed that LS helped to increase the solution search space of the IG algorithm.

In addition to improving the performance of the IG, other reasons behind using BR-IG-LS heuristic to solve deterministic RSP is due to the fact that it is easy to implement, parameter free and suitable for large-size problems. Moreover, the LS can improve the search mechanism on the construction phase of the IG algorithm by increasing size of the search space, improving speed, and providing better results in the solution generation procedure.

To the best of our knowledge, our solution method is the first to use both biased randomised and local search to improve the IG algorithm for RSPs with disruptions.

4.3. Proposed Biased Randomised IG with Local Search (BR-IG-LS) for solving RSP

We propose the IG-BR algorithm and add the LS (Algorithm 4.1) to improve the quality of the solution. The pseudo-code of the basic IG-BR algorithm is presented as Algorithm 3.2.

The main process of LS is moving in a search space to find the local optimal by searching on neighbourhood of the current solution. Let S' be the current solution and $N(S')$ present the set of solutions that can be achieved. LS attempts to select a new solution from the neighbourhood of $N(S')$ using the concept of insertion neighbourhood structure. In order to do that, LS evaluates the neighbourhood solution, if the new solution (S) has an improvement, it will be saved as the new current solution (S') and we continue to search to find a better neighbourhood solution. Search will continue until no better neighbourhood solution is found , which means that we reached to a local optimal (Johnson et al., 1988).

Algorithm 4.1 below describes a basic local search heuristic procedure for a minimisation problem; the basic concept of local search is started by setting an initial feasible solution (Pseudocode: line 2), then finding the new neighbourhood solution around the initial solution (Pseudocode: line 4). In this thesis, we focus on the objective to minimise total train delay. Therefore, if the value of the new solution *V(S)* is lower than or equal to the value of the current solution *V(S')* (Pseudocode: line 5), replace the current solution *S* with the initial solution *S'* (Pseudocode: line 6). Then, repeat the process until the time limit is reached to find the local optimal (Orlin et al., 2003).

Algorithm 4.1 : Procedure for basic Local Search Algorithm (Orlin et al., 2003)

1: **Procedure basic Local Search Algorithm**
2: *S* ': *GenerateInitialFeasible*(); ⊳ *Initial Solution*
3:     **while** *S* ' is not local optimal **do**
4:     find a new neighbourhood solution *S* ∈ *N(S ')* ⊳ *using insertion neighbourhood structure*
5:         **if** *V(S)* ≤ *V(S')* **then**
6:             set *S'* = *S*
7:         **end if**
8:     **end while**
9: **end procedure**

Some researchers suggested applying LS to improve the IG in the literature as follows Task assignment (Harish et al., 2014), Task allocation (Kang et al., 2013), Parallel machine scheduling (Ying et al., 2010), Freight train scheduling (Yuan et al., 2008) and Flow shop scheduling (Ruiz et al., 2008). The literature review showed that LS can improve the efficiency of IG algorithms and the LS concept that is mostly combined with the IG algorithm is the insertion neighbourhood concept. Thus, it is reasonable to combine LS techniques with IG algorithm to solve RSP.

Figure 4.1 shows how to investigate a local search concept in the IG-BR algorithm which shown in big **red** arrow. On the BR-IG-LS algorithm for RSP, the process starts by working on the destruction phase; we add the biased randomised algorithm to the end of the destruction phase for reordering the construction set before inserting back into the remaining set. After that, working on the construction phase, the local search algorithm is applied to the construction phase by reinserting the first element in a construction set back into the remaining set randomly. Then, repeat the whole process again until the stopping criteria are met. Moreover, the LS concept that we used in our proposed method is also the insertion neighbourhood concept.

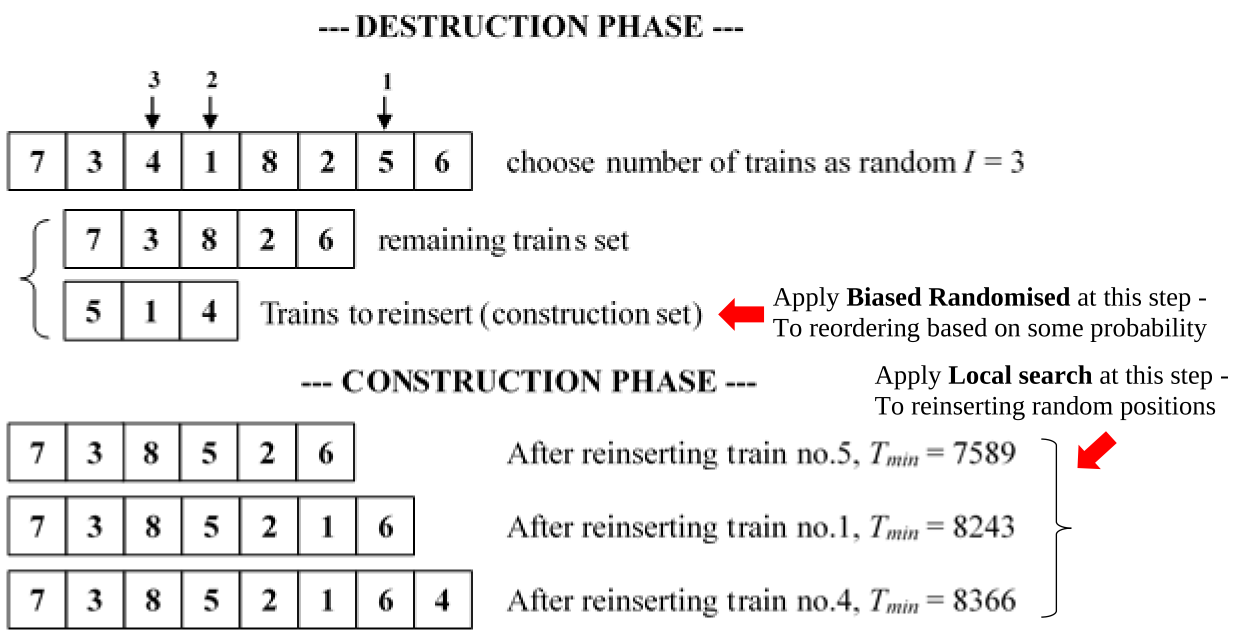

Figure 4.1 : Apply local search to IG biased randomised (Ruiz et al., 2007)

We propose an extension by applying LS into the IG-BR algorithm in order to improve the solution of the deterministic RSP under the uncertainties delay. The main procedure is described as follows: (a) Initial solution of timetable as our case studies from the UK and Thailand Rail networks, (b) IG is used to solve the RSP by using two main phases (destruction and construction), (c) use of biased randomised methods to re-order a construction set, (d) the use of local search methods to improve the speed of solution (Algorithm 4.2).

Algorithm 4.2 : Procedure for Deterministic RSP BR-IG-LS Algorithm

---

1: **Procedure for Deterministic RSP BR-IG-LS Algorithm** (r', r^D, r^R, d, b, β)
▻ r': Initial solution
▻ r^D: Partial sequence to reconstruct
▻ r^R: Trains to reinsert
▻ d: Randomly chosen number of trains
▻ b: Number of blocks
▻ β: Parameter for biased randomised
2: r': *GenerateInitialSolution*(); ▻ *Initial solution*
3: **while** execution_time < 400 **do** ▻ *Execution time 400 seconds*
4: **for** i = 1 to b **do** ▻ *Run for all block*
5: set r^R = empty
6: **for** i = 1 to d **do** ▻ *Destruction step*
7: r^R <- remove one node randomly from r' and insert it in r^R;
8: $r^D = r'$ <- the remaining set of trains
9: **end for**
10: re-ordering r^R by using β to assign a probability ▻ *Biased randomised*
11: **for** j = 1 to d **do** ▻ *Construction step*
12: r^{ans} = ApplyLocalSearch(r^R) ▻ *Insertion Neighbourhood (LS)*
13: **if** r^{ans} met with all railway constraints **then** ▻ *check all constraints*
14: solution (r^{ans}) = calculate minimise a total delay (r^{ans})
15: **end if**
16: **end for**
17: **end for**
18: **if** solution (r^{ans}) < solution (r') **then** ▻ *Minimise the total delay*
19: solution (r^{best}) = solution (r^{best});
20: **else if**
21: solution (r^{best}) = solution (r');
22: **end if**
23: **end while**
24: **end procedure**

Algorithm 4.2 presents the pseudocode for deterministic RSP BR-IG-LS Algorithm. Firstly, the process starts by creating an initial solution from the rail timetable (Pseudocode: line 2). We consider only the deterministic disruptions, so the amount of the delay is known beforehand and we input it together with the existing timetable.

Then, we start to implement the destruction phase of IG to find a new feasible solution (Pseudocode: line 6 to line 9). However, at the end of the destruction phase, we apply the biased randomised algorithm to reorder the construction set by using the geometric probability distribution, which can give probabilities to each train in the construction set. Trains with smaller running time are more likely to be selected than the other ones with large running times before continuing to the next step (Pseudocode: line 10).

Moreover, at the beginning of the construction phase, we reinsert the construction set back into the remaining set by using the concept of local search with insertion neighbourhood structure. This step is reinserting in some random positions and to find the best sequence that provides the local optimal solution (Pseudocode: line 11 to line 16). In this step, we need to confirm that all railway operational constraints are met (Pseudocode: line 13 to line 15).

Following, the destruction phase, construction phase, biased randomised phase and the local search phase, we use simplest acceptance criteria to check that a new timetable has provided a better solution or not, as smallest total delay and set it as the current solution for next iteration (Pseudocode: line 18 to line 22). After that, the new schedule will be sent to the rail control centre for management of the rail process. Finally, we repeat all processes until the stopping criteria of 400 seconds are met.

4.4. Computational experiments

In this section, we evaluate the performance of the BR-IG-LS, which is the proposed solution method in this chapter by comparing the results from the previous chapter. The same set of data from Chapter 3 was used, consisting of two different cases which are the UK and Thailand Rail network. All constraints are discussed in section 3.5.

The major aim of conducting experiments is to find the minimum total train delays. BR-IG-LS was implemented on a personal computer using an Intel core i7-4500U CPU, 1.80GHz-2.4GHz, and 8GB RAM.

4.4.1. Southeastern train company, United Kingdom

The BR-IG-LS was compared to the result from the previous chapter with the same computation time of 400 seconds. In this chapter, we stopped the CPLEX at 400 seconds, because the CPLEX using more than 7 hours to find the optimal solution for the test instances. Therefore, comparison between CPLEX, IG-BR and BR-IG-LS was used to evaluate the performance of the BR-IG-LS.

Tables 4.1-4.4 show the solution of using CPLEX, IG-BR and BR-IG-LS to solve the UK case study. In each table, all columns before the CPLEX column show the detail of each instance and all delay are shown in **minutes**. The other columns present results of each solution method and their performance. Finally, at the end of each table, we calculated an average of the total delay and RD (%).

Block Delay	# of Block(s)	# of Train(s)	Instance	CPLEX	IG-BR		BR-IG-LS		
				Delay	Delay	RD (%) compared with CPLEX	Delay	RD (%) compared with IG-BR	RD (%) compared with CPLEX
5	1	3	P01	107.83	97.67	9.42	**95.33**	**2.40**	**11.59**
	2	5	P02	143.67	**140.33**	**2.32**	150.50	-7.25	-4.76
10	1	3	P03	167.00	168.67	-1.00	**160.83**	**4.65**	**3.69**
	2	5	P04	**281.83**	314.83	-11.71	295.17	6.24	-4.73
15	1	3	P05	260.17	**247.33**	**4.93**	**247.33**	**0.00**	**4.93**
	2	5	P06	314.67	313.33	0.42	**311.67**	**0.53**	**0.95**
20	1	9	P07	**1028.00**	1099.83	-6.99	1073.50	2.39	-4.43
	2	12	P08	**661.75**	705.50	-6.61	731.00	-3.61	-10.46
25	1	9	P09	**1272.67**	1471.17	-15.60	1308.83	11.03	-2.84
	2	12	P10	1823.00	1829.00	-0.33	**1816.67**	**0.67**	**0.35**
30	1	9	P11	**1021.00**	1023.33	-0.23	1023.33	0.00	-0.23
	2	12	P12	**1769.58**	2052.00	-15.96	2051.83	0.01	-15.95
Avg Delay				**737.60**	788.58	-6.91	772.17	**2.08**	-4.69

Table 4.1 : Deterministic Longer running/dwell times - minor and general disruption: BR-IG-LS Results

Table 4.1 shows the BR-IG-LS results of deterministic RSP on minor and general delays on longer running time and dwell time. The best solution for each instance is indicated in **bold**. We set the limit of the computation time to 400 seconds which stopped the CPLEX reaching the optimal solution. The overall average of total train delays showed that CPLEX also provided the best efficiency for this type of disruption. A comparison between CPLEX, IG-BR and BR-IG-LS showed that CPLEX reduced the average of total train delays by 6.91% more than IG-BR and 4.69% than BR-IG-LS. Moreover, if we only focus between IG-BR and BR-IG-LS which is the approximate algorithm, the results showed that BR-IG-LS reduced the total delay time by 2.08%.

Late Departure	# of Train(s)	Instance	CPLEX	IG-BR		BR-IG-LS		
			Delay	Delay	RD (%) compared with CPLEX	Delay	RD (%) compared with IG-BR	RD (%) compared with CPLEX
5	1	R01	**61.83**	70.83	-14.55	68.67	3.05	-11.06
	3	R02	**59.83**	65.50	-9.47	62.83	4.08	-5.01
	5	R03	**73.17**	75.17	-2.74	80.33	-6.86	-9.79
10	1	R04	**70.67**	78.50	-11.09	75.00	4.46	-6.13
	3	R05	**82.17**	83.50	-1.62	82.83	0.80	-0.80
	5	R06	**138.00**	143.83	-4.22	143.17	0.46	-3.75
15	1	R07	**67.66**	68.67	-1.49	68.67	0.00	-1.49
	3	R08	**119.33**	127.33	-6.70	124.83	1.96	-4.61
	5	R09	**167.83**	171.33	-2.08	171.33	0.00	-2.08
20	6	R10	**182.33**	184.00	-0.92	183.67	0.18	-0.73
	9	R11	**240.50**	250.67	-4.23	246.83	1.53	-2.63
	12	R12	**334.17**	347.67	-4.04	347.67	0.00	-4.04
25	6	R13	**230.67**	240.33	-4.19	233.00	3.05	-1.01
	9	R14	**275.67**	281.67	-2.18	280.00	0.59	-1.57
	12	R15	**376.67**	385.17	-2.26	381.33	1.00	-1.24
Avg Delay			**165.37**	171.61	-3.78	170.01	**0.93**	-2.81

Table 4.2 : Deterministic Late departures - minor and general disruption: BR-IG-LS Results

Table 4.2 shows the BR-IG-LS results of deterministic RSP on minor and general delays on late departure time. The best solution for each instance is indicated in **bold**. All experimentations were based on 400 seconds. CPLEX exceeded this time limit and did not achieve the optimal solution. The experimental results clearly showed that CPLEX gave a better performance to decrease average total delay of all trains than the other algorithms by 3.78% more than IG-BR and 2.81% more than BR-IG-LS. However, if we only focused on both types of IG algorithms, the performance of BR-IG-LS slightly reduced the train delay by 0.93% lower than IG-BR.

Block Delay	# of Block(s)	Instance	CPLEX	IG-BR		BR-IG-LS		
			Delay	Delay	RD (%) compared with CPLEX	Delay	RD (%) compared with IG-BR	RD (%) compared with CPLEX
40	4	HP01	7706.00	7379.33	4.24	**7227.47**	**2.06**	**6.21**
	5	HP02	7432.00	7260.75	2.30	**7118.82**	**1.95**	**4.21**
	6	HP03	7619.50	**7250.47**	**4.84**	7335.30	-1.17	3.73
50	4	HP04	9396.67	8836.87	5.96	**8180.50**	**7.43**	**12.94**
	5	HP05	9342.33	9150.20	2.06	**8997.90**	**1.66**	**3.69**
	6	HP06	8529.00	7548.50	11.50	**7494.00**	**0.72**	**12.14**
60	4	HP07	11574.83	10429.50	9.90	**9987.60**	4.24	**13.71**
	5	HP08	9309.58	**8795.90**	**5.52**	8871.20	-0.86	4.71
	6	HP09	9570.67	**8497.00**	**11.22**	**8497.00**	**0.00**	**11.22**
70	4	HP10	13182.67	12187.30	7.55	**11879.10**	**2.53**	**9.89**
	5	HP11	10716.58	9656.60	9.89	**9575.33**	**0.84**	**10.65**
	6	HP12	13006.42	11795.70	9.31	**11685.17**	**0.94**	**10.16**
80	4	HP13	15491.00	14262.90	7.93	**14151.60**	**0.78**	**8.65**
	5	HP14	14858.67	13645.60	8.16	**13320.50**	**2.38**	**10.35**
	6	HP15	15646.83	14103.50	9.86	**14052.80**	**0.36**	**10.19**
Avg Delay			10892.18	10053.34	7.70	**9891.62**	**1.61**	**9.19**

Table 4.3 : Deterministic Longer running/dwell times - major disruption: BR-IG-LS Results

Table 4.3 presents the BR-IG-LS results of deterministic RSP on major delays on longer running time and dwell times. The best solution for each instance is indicated in **bold**. The best results provided by BR-IG-LS which gave the highest performance in this set of data. On the overall average of total train delay, BR-IG-LS was better than CPLEX by 9.19% and better than IG-BR by 1.61%. Moreover, we could not confirm that BR-IG-LS was better than CPLEX because in this case study CPLEX execution was stopped at 400 seconds.

Block Delay	# of Block(s)	Departure Delay	# of Train(s)	Instance	CPLEX	IG-BR		BR-IG-LS		
					Delay	Delay	RD (%) compared with CPLEX	Delay	RD (%) compared with IG-BR	RD (%) compared with CPLEX
40	4	10	3	PR01	7524.00	7379.92	1.91	**7347.67**	**0.44**	**2.34**
			5	PR02	7455.67	**7285.42**	**2.28**	7311.70	-0.36	1.93
		20	9	PR03	7599.17	6975.83	8.20	**6884.33**	**1.31**	**9.41**
			12	PR04	7823.50	7107.00	9.16	**7017.20**	**1.26**	**10.31**
40	6	10	3	PR05	6830.00	6930.50	-1.47	**6816.00**	**1.65**	**0.20**
			5	PR06	7773.56	7002.83	9.91	**6997.75**	**0.07**	**9.98**
		20	9	PR07	6518.50	**6465.85**	**0.81**	6515.83	-0.77	0.04
			12	PR08	8088.50	7458.50	7.79	**7333.67**	**1.67**	**9.33**
50	4	10	3	PR09	9462.17	9346.13	1.23	**9212.50**	**1.43**	**2.64**
			5	PR10	9615.75	9097.40	5.39	**8849.17**	**2.73**	**7.97**
		20	9	PR11	9589.00	**7672.54**	**19.99**	**7672.54**	**0.00**	**19.99**
			12	PR12	9863.33	9194.50	6.78	**9087.30**	**1.17**	**7.87**
50	6	10	3	PR13	9346.42	8400.54	10.12	**8367.80**	**0.39**	**10.47**
			5	PR14	9737.17	**9741.20**	**-0.04**	**9741.20**	**0.00**	**-0.04**
		20	9	PR15	8609.00	8529.00	0.93	**8432.40**	**1.13**	**2.05**
			12	PR16	8286.92	8012.83	3.31	**7967.08**	**0.57**	**3.86**
Avg Delay					8382.67	7912.50	5.61	**7847.13**	**0.83**	**6.39**

Table 4.4 : Deterministic Longer running/dwell times and Late departures - major disruption: BR-IG-LS Results

Table 4.4 indicates BR-IG-LS results of deterministic RSP for major disruption on late departure times. The best solution for each instance is indicated in **bold**. To compare the results based on the same execution time of 400 seconds, the average total train delays of BR-IG-LS were 0.83% lower than IG-BR and 6.39% lower than CPLEX.

The SSDP separated the disruptions into 3 types depending on the size of disruption as minor disruptions, general disruptions and major disruptions. Then, we discussed our results based on the disruption types.

Disruption Type		CPLEX	IG-BR		BR-IG-LS		
		Delay	Delay	RD (%) compare with CPLEX	Delay	RD (%) compare with IG-BR	RD (%) compare with CPLEX
Minor and General	Longer Running/dwell times	**737.60**	788.58	-6.91	772.17	2.08	-4.69
	Late Departures	**165.37**	171.61	-3.78	170.01	0.93	-2.81
Average Delay		**451.48**	480.10	-6.34	471.09	**1.88**	-4.34

Table 4.5 : Summary of Experimental result of BR-IG-LS - Deterministic (UK) - Small

Table 4.5 shows the summary results from Tables 4.1 and 4.2 considered on small disruption instances. When we computed CPLEX, IG-BR and BR-IG-LS for only 400 seconds, the best solution was provided by CPLEX which was 6.34% better than IG-BR and 4.34% better than BR-IG-LS for reducing the total train delays. As we know from the literature, CPLEX is more suitable for small case studies because it needs huge amounts of computation time to solve complex RSPs. However, when only focusing on both types of

IG algorithms, the performance of BR-IG-LS can reduce train delay by 1.88% lower than IG-BR.

Disruption Type		CPLEX	IG-BR		BR-IG-LS		
		Delay	Delay	RD (%) compare with CPLEX	Delay	RD (%) compare with IG-BR	RD (%) compare with CPLEX
Major	Longer Running/dwell times	9637.42	8982.92	6.79	**8869.38**	1.26	7.97
	Late Departures & Longer Running/dwell times	5044.45	4731.51	6.20	**4670.23**	1.30	7.42
Average Delay		7340.94	6857.21	6.59	**6769.80**	**1.27**	**7.78**

Table 4.6 : Summary of Experimental result of BR-IG-LS - Deterministic (UK) - Large

Table 4.6 shows the summary results from Tables 4.3 and 4.4 considered on large disruption instances. After stopping the implementation at 400 seconds, the results showed that BR-IG-LS was more effective at reducing the average total train delay when compared with the IG-BR at 1.27% and CPLEX at 7.78%. As expected, CPLEX cannot provide good performance in large size of disruption because it needs high computation time that exceeded the limit in our case study.

Disruption Type		CPLEX	IG-BR		BR-IG-LS		
		Delay	Delay	RD (%) compare with CPLEX	Delay	RD (%) compare with IG-BR	RD (%) compare with CPLEX
Minor and General	Longer Running/dwell times	**737.60**	788.58	-6.91	772.17	2.08	-4.69
	Late Departures	**165.37**	171.61	-3.78	170.01	0.93	-2.81
Major	Longer Running/dwell times	10892.18	10053.34	7.70	**9891.62**	1.61	9.19
	Late Departures & Longer Running/dwell times	8382.67	7912.50	5.61	**7847.13**	0.83	6.39
Average Delay		5044.45	4731.51	6.20	**4670.23**	**1.30**	**7.42**

Table 4.7 : Summary of Experimental result of BR-IG-LS - Deterministic (UK) - All

Table 4.7 presents a summary of all results in section 4.4.1.1 and shows that CPLEX was effective for solving minor and general disruption RSP. However, BR-IG-LS was the best choice for the major disruption RSP and decreased the total delay from IG-BR solution by 1.30%. To compare BR-IG-LS with the solution executed for 400 seconds from CPLEX, resulted in 7.42% gap of the average total train delays.

In addition, if we only considered the two algorithms of IG-BR with and without LS, the overall results in this chapter showed that BR-IG-LS had better performance by 1.30%, which means that LS can improve the efficiency of IG-BR.

4.4.2. State Railway of Thailand

The computational experiments comparing between historical data, CPLEX, IG-BR and BR-IG-LS are shown in this chapter. All experiments were based on 400 seconds time limit to generate a new schedule. We cannot run CPLEX to receive the optimal solution, because it takes more than 15 hours to solve the Thailand case study. Therefore, we compared all solution methods based on 400 seconds time frame to evaluate the performance of BR-IG-LS.

Table 4.8 shows the experimental results using historical data, CPLEX, IG-BR and BR-IG-LS. In this table, the first two columns show the detail of each instance and all delays data were provided in **minutes**. The historical data column showed total delay of all trains provided by the Thailand rail company. After that, CPLEX and IG-BR solutions from the previous chapter were provided in the next two columns. Then, column BR-IG-LS showed the total delay of each instance and the improvement when compared to the other methods from Chapter 3 by using RD (%). Finally, at the end of the table, we calculated an average of the total delay and RD.

Date	Instance	Historical Data	CPLEX	IG-BR	BR-IG-LS			
		Delay	Delay	Delay	Delay	RD (%) compared with historical data	RD (%) compared with IG-BR	RD (%) compared with CPLEX
1/2/2019	THAF101	2477	2121	1973	**1937**	21.80	1.82	8.68
2/2/2019	THAF102	2138	2035	1998	**1958**	8.42	2.00	3.78
3/2/2019	THAF103	1830	1803	1789	**1755**	4.10	1.90	2.66
4/2/2019	THAF104	1681	1325	**1154**	1187	29.39	-2.86	10.42
5/2/2019	THAF105	2549	2379	2247	**2206**	13.46	1.82	7.27
6/2/2019	THAF106	1750	1258	1234	**1198**	31.54	2.92	4.77
7/2/2019	THAF107	2329	2188	2107	**2093**	10.13	0.66	4.34
8/2/2019	THAF108	3186	2648	**2422**	2434	23.60	-0.50	8.08
9/2/2019	THAF109	2143	1907	1895	**1822**	14.98	3.85	4.46
10/2/2019	THAF110	2180	1574	1535	**1509**	30.78	1.69	4.13
11/2/2019	THAF111	2712	1982	1825	**1782**	34.29	2.36	10.09
12/2/2019	THAF112	2472	1830	1794	**1742**	29.53	2.90	4.81
13/2/2019	THAF113	2440	1874	**1713**	**1713**	29.80	0.00	8.59
14/2/2019	THAF114	1807	1807	1780	**1704**	5.70	4.27	5.70
15/2/2019	THAF115	2903	2458	**2265**	2268	21.87	-0.13	7.73
16/2/2019	THAF116	3582	3030	2877	**2655**	25.88	7.72	12.38
17/2/2019	THAF117	3516	2740	2554	**2422**	31.11	5.17	11.61
18/2/2019	THAF118	2895	2645	2321	**2278**	21.31	1.85	13.88
19/2/2019	THAF119	1880	1457	1398	**1328**	29.36	5.01	8.85
20/2/2019	THAF120	2791	1861	1801	**1715**	38.55	4.78	7.85
21/2/2019	THAF121	3798	3050	**2821**	2845	25.09	-0.85	6.72
22/2/2019	THAF122	3658	3421	3293	**3287**	10.14	0.18	3.92
23/2/2019	THAF123	3772	3543	3355	**3317**	12.06	1.13	6.38
24/2/2019	THAF124	3597	3260	3180	**3034**	15.65	4.59	6.93
25/2/2019	THAF125	3974	3477	**3321**	3358	15.50	-1.11	3.42
26/2/2019	THAF126	4873	4495	4288	**4251**	12.76	0.86	5.43
27/2/2019	THAF127	3385	3392	3275	**3180**	6.06	2.90	6.25
28/2/2019	THAF128	3023	2584	2447	**2331**	22.89	4.74	9.79
Average Delay		2833.61	2433.71	2309.36	**2261.04**	**20.21**	**2.09**	**7.10**

Table 4.8 : Deterministic Thai – BR-IG-LS Results

Table 4.8 presents the BR-IG-LS results of deterministic RSP on the Thailand case study. The best solution for each instance is indicated in **bold**. The results show that when we applied LS to improve the IG-BR solution on the real-world data from Thailand, it provided an improvement by reducing the average total train delay by 2.09%. Moreover, if we consider the company's current solution, it shows that BR-IG-LS increase the efficiency of the rail management process by 20.21%, decreasing the total train delays. However, the CPLEX did not show a good performance in this case study, with the total delay worse than BR-IG-LS by 7.10%, because we used limited computation time as 400 seconds and the program needed large amounts of computation time to solve the Thailand case study.

4.5. Chapter Conclusion

The major aims in this chapter were to improve the quality of IG-BR from the previous chapter by added LS to increase the size of the search space, called BR-IG-LS. The LS satisfied the aim of improving the search procedure, so it is reasonable to combine LS techniques with IG-BR to tackle the RSP. In this chapter, we also used the optimisation model with the objective to minimise the total train delays.

In the procedure of BR-IG-LS algorithm, trains are randomly selected out of the sequence in the destruction phase and reordered using biased randomisation. Then, trains were reinserted in the construction phase by applying the LS with neighbourhood search to move from one candidate to another in a search space.

We compared the proposed method with the result of the real-world case studies from the Southeastern train company, UK (Khosravi, 2013) and State Railway of Thailand from Chapter 3. The computation time limit was 400 seconds for all instances. In the UK test instances, the overall results showed that BR-IG-LS was outstanding for the deterministic RSP; however, on the minor and general disruption set of instances, the best solution was provided by CPLEX. For the Thailand case study, the experimental results showed that the BR-IG-LS was more effective when compared with the company's historical solution and IG-BR. Therefore, the BR-IG-LS was more productive for solving RSP in large and complex problems within the limit of computation time.

Results in this chapter showed a large gap between BR-IG-LS and CPLEX in the minor and general disruption, with the opportunity to improve the quality of solution. However, the BR-IG-LS algorithm reached a limit and was hard to improve by combination with the other techniques. Therefore, we need other methods to solve the deterministic RSP. For this reason, in the next chapter, we developed a combination of VNS and a biased randomised algorithm to generate and find a better solution for RSP.

Chapter 5: Biased Randomised Variables Neighbourhood Search (BR-VNS) for deterministic RSP

5.1. Introduction

The results in the previous chapter showed that BR-IG-LS algorithm had good efficiency when solving the deterministic RSP. However, there was a large gap between BR-IG-LS and CPLEX in the minor and general disruption, so we proposed another method to improve the quality of solution. We selected a new metaheuristic method, namely Variable Neighbourhood Search (VNS) which had been successfully used in other COPs to solve the deterministic RSP.

In this chapter, we proposed a BR-VNS algorithm to solve the RSP that deals with a deterministic delay. This chapter discusses the basic concept of VNS which is a new metaheuristic method introduced by Hansen et al. (2001) and widely studied to solve COPs in the last two decades. Samà et al. (2017) were the first to propose the VNS algorithm to solve RSP with effective results as a similar problem of RSP. Moreover, the parallel machines job shop scheduling problem which is very similar to our optimisation model was successfully solved by VNS (Driessel et al., 2011). According to the literature review and results from Chapter 3, biased randomisation can help to improve the main algorithm. Therefore, in this chapter, we proposed the BR-VNS algorithm to solve the RSP that deals with a deterministic delay.

This chapter mainly discusses the concept of BR-VNS and then shows how to use BR-VNS to solve a deterministic RSP. After that, experimental results of the UK and Thailand case studies are presented to evaluate the performance of the proposed algorithm.

The chapter is structured as follows: section 5.2 presents this chapter's contributions, then the proposed BR-VNS algorithm to solve RSP with deterministic disruption delays is presented in section 5.3. All the experimental results are shown in section 5.4, separate into two sets of case studies (UK & Thailand). Section 5.5 presents the conclusion of this chapter.

5.2. Contribution

The VNS has been very successful in solving different types of optimisation problems such as Capacitated location routing (Derbel et al., 2011), Job shop scheduling (Liao et al., 2007; Roshanaei et al., 2009; Zandieh et al., 2010) and Vehicle routing (Bräysy, 1999; Polacek et al., 2005). Samà et al. (2017) proposed the VNS algorithm to solve a train scheduling and routing problem and their results outperform Tabu Search and MILP methods (CPLEX). As VNS has been adapted successfully on many COPs, it was reasonable to apply VNS to combine with other techniques to improve the solution of the VNS algorithm. Juan et al. (2013) claimed that biased randomisation can add diversification and increase performance of the main algorithm; therefore, we selected this biased randomised heuristic to improve the VNS algorithm. To the best of our knowledge, the BR-VNS has not been used in the RSP before, therefore this is the first study to employ biased randomisation to improve the VNS algorithm for the RSP that deal with delays.

In addition, other reasons behind using BR-VNS heuristic to solve deterministic RSP are because the BR-VNS algorithm is easy to implement and can provide randomised behaviour for the basic VNS algorithm.

5.3. Proposed Biased Randomised VNS for solving RSP

The VNS is a metaheuristic which consists of two or more neighbourhood structures to find the solution for many COPs. The basic steps of VNS are shown in Figure 5.1 below (Hansen et al., 2001).

Initialization. Select the set of neighborhood structures $\mathcal{N}_k$, $k = 1, \dots, k_{max}$, that will be used in the search; find an initial solution x; choose a stopping condition;

Repeat the following until the stopping condition is met:
(1) Set $k \leftarrow 1$; (2) Until $k = k_{max}$, repeat the following steps:
(a) *Shaking.* Generate a point x' at random from the k^{th} neighborhood of x ($x' \in \mathcal{N}_k(x)$);
(b) *Local search.* Apply some local search method with x' as initial solution; denote with x'' the so obtained local optimum;
(c) *Move or not.* If this local optimum is better than the incumbent, move there ($x \leftarrow x''$), and continue the search with $\mathcal{N}_1$ ($k \leftarrow 1$); otherwise, set $k \leftarrow k + 1$;

Figure 5.1 : Steps of basic VNS (Hansen et al., 2001)

Algorithm 5.1 shows how the basic VNS works in detail. We denote N_k, $k = 1, ..., k^{max}$ as a set of neighbourhood structures to solve the problem and $N_k(x)$ is a set of solutions in the k^{th} neighbourhood of x, where x refers to an initial solution (Pseudocode: line 2 to line 3). Firstly, the VNS process starts with the first neighbourhood structure N_k, (k=1) (Pseudocode: line 6) and generates a random start point x' from each k^{th} neighbourhood of x in order to avoid cycling (Shaking phase) (Pseudocode: line 7). Secondly, find a new local optimal solution x'' around the solution of shaking phase by using the k^{th} neighbourhood rule (Local search phase) (Pseudocode: line 8). Then, we check if the new solution (x'') is better than best solution (x), set $x = x''$ and continue the search with N_k, (k=1); otherwise move to use the next neighbourhood structure N_k, (k=k+1) (Move or not phase) (Pseudocode: line 9 to line 14). Finally, repeat the process until the stopping condition is met, or finish on the last neighbourhood structure N_k, ($k \geq k^{max}$) (Pseudocode: line 5 to line 16).

Algorithm 5.1 : Procedure for basic VNS Algorithm

1: **Procedure VNS Algorithm** (x) ⊳ x : Initial solution
2: set of neighbourhood structures N^k; k = 1, ..., k^{max} ⊳ *set neighbourhood structures*
3: x : $GenerateInitialSolution()$; ⊳ *Initial Solution*
4: $k = 1$
5: **while** stopping criteria are not met **do**
6: **for** $k = 1$ to k^{max} **do**
7: select a random x' from $N_k(x)$ ⊳ *shaking*
8: x'' = neighbourhood search of x' ⊳ *local search*
9: **if** $x'' < x$ **then** ⊳ *move or not*
10: $x = x''$
11: $k = 1$
12: **else if**
13: $k = k + 1$
14: **end if**
15: **end for**
16: **end while**
17: **end procedure**

We describe the main procedure of basic VNS to solve the deterministic RSP as follows: (a) select the set of neighbourhood structures, which consider the rail network characteristics (b) initial solution of timetable, which is collected from the rail public website and (c) VNS, which improves the solution by using three main phases (shaking, local search and move or not) (Algorithm 5.2).

For the set of neighbourhood structures, two types of neighbourhood structure mostly used in the scheduling problem are insertion and swap (Liao et al., 2007); we also use these popular structures to create the 4 neighbourhood structures in our VNS as below:

- k=1; Swap two train positions on the same blocks. Randomly select two trains which operate on the same block, then place each train back into the position previously occupied by the other.

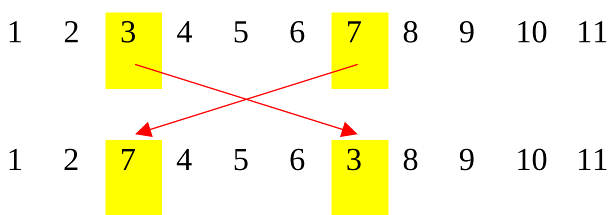

Figure 5.2 : First neighbourhood structure (k=1)

- k=2; Swap three train positions on the same block. Randomly select three trains which operate on the same block, then randomly place each train back into the position previously occupied by the others.

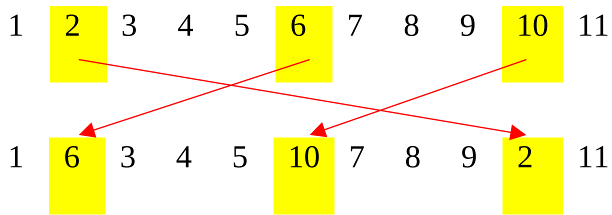

Figure 5.3 : Second neighbourhood structure (k=2)

- k=3; Insert one train. Randomly select one train from the sequence, then reinsert it back in a random position on the same block.

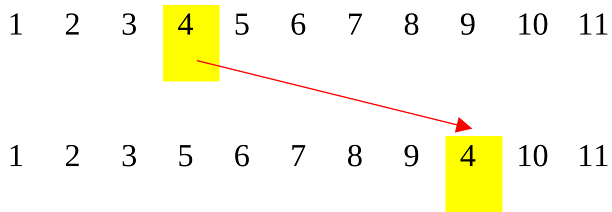

Figure 5.4 : Third neighbourhood structure (k=3)

- k=4; Insert two trains. Randomly select two trains from the sequence, then reinsert those trains back in random positions on the same block.

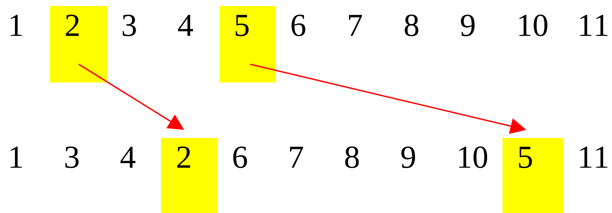

Figure 5.5 : Fourth neighbourhood structure (k=4)

Algorithm 5.2 describes how to use basic VNS to solve the deterministic RSP. Firstly, the process starts by selecting the set of neighbourhood structures as shown in Figures 5.2-5.5 (Pseudocode: line 2). Secondly, we create an initial solution from the rail timetable (Pseudocode: line 3).

The process of VNS for deterministic RSP starts from the first neighbourhood structure until the last neighbourhood structure (Pseudocode: line 6). In each neighbourhood, we need to perform the shaking phase, local search phase and move or not phase.

Shaking phase - we randomly select an *x'* solution from the initial solution *x* in order to avoid cycling (Pseudocode: line 7) and check that the solution matches with all operational constraints such as departure time, running/dwell time and headway constraints (Pseudocode: line 8 to line 10).

Local search phase - we use the k^{th} neighbourhood rule to find a local optimal (Pseudocode: line 11) and also need to recheck that all operational constraints are still valid (Pseudocode: line 13 to line 14).

Move or not phase - we check if the new solution *x''* is better than the best solution *x* or not, if yes set best solution *x* = new solution *x''* and continue to search on the first neighbourhood structure N_k, (k=1); otherwise move to use the next neighbourhood structure N_k, (k=k+1) (Pseudocode: line 15 to line 20).

Algorithm 5.2 : Procedure for Deterministic RSP VNS Algorithm

```
-----------------------------------------------------------------------------------------------
1: Procedure Deterministic RSP based VNS Algorithm (x)
                                                        ▻ x: Initial solution
2: set of neighbourhood structures N^k; k = 1, ..., k^max  ▻ set neighbourhood structures
3: x: GenerateInitialSolution();                    ▻ Initial Solution
4: k = 1                                                ▻ start from first neighbourhood structure
5: while execution_time < 400 do                        ▻ Execution time 400 seconds
6:    for k = 1 to k^max do                             ▻ k^max=4; four neighbourhood structures
7:         select a random x' from N_k(x)               ▻ shaking
8:         if x' met with all railway constraints then  ▻ check all constraints
9:             solution (x') = calculate minimise a total delay (x')
10:        end if
11:            x'' = neighbourhood search of x'         ▻ local search
12:        if x'' met with all railway constraints then ▻ check all constraints
13:            solution (x'') = calculate minimise a total delay (x'')
14:        end if
15:        if solution (x'') < solution (x) then        ▻ move or not
16:            x = x''
17:            k = 1                                    ▻ return to first neighbourhood structure
18:        else if
19:            k = k + 1                                ▻ move to next neighbourhood structure
20:        end if
21:   end for
22: end while
23:end procedure
```

5.3.1. Biased Randomised VNS (BR-VNS)

The VNS algorithm to solve a train scheduling and routing problem was performed by Samà et al. (2017). Their results were very successful. Moreover, we know that biased randomisation can improve the performance of algorithms by avoiding the local optimal and increase diversification of the heuristics or metaheuristics. However, the VNS algorithm has not yet been combined with biased randomisation technique. Thus, it is reasonable to combine biased randomisation techniques with the basic VNS algorithm to solve the deterministic RSP, called BR-VNS.

The BR-VNS uses a biased randomised algorithm concept to generate an initial solution and then starts the VNS algorithm to improve the solution. The process of BR-VNS implementation is described as follows: (a) select the set of neighbourhood structures which match to the rail network characteristics (b) generate an initial solution by using a biased randomised algorithm, and (c) VNS, which improves the solution following three main phases (shaking, local search and move or not) (Algorithm 5.3).

Algorithm 5.3 : Procedure for Deterministic RSP BR-VNS Algorithm

```
-----------------------------------------------------------------------------------------
1: Procedure Deterministic RSP BR-VNS Algorithm (x, b, β)
                                                        ⊳ x: Initial solution
                                                        ⊳ b: Number of blocks
                                                        ⊳ β: Parameter: biased randomised
2: set of neighbourhood structures N^k; k = 1, ..., k^max      ⊳ set neighbourhood structures
3: for i = 1 to b do                                           ⊳ run for all block
4:    using β probability to assign order of train in each block   ⊳ Biased randomised
5: end for
6: x: GenerateBiasedRandomisedInitialSolution();        ⊳ Initial solution
7: k = 1                                                ⊳ start from first neighbourhood structure
8: while execution_time < 400 do                        ⊳ Execution time 400 seconds
9:    for k = 1 to k^max do                             ⊳ k^max=4; four neighbourhood structures
10:       select a random x' from N_k(x)                ⊳ shaking
11:       if x' met with all railway constraints then   ⊳ check all constraints
12:          solution (x') = calculate minimise a total delay (x')
13:       end if
14:          x" = neighbourhood search of x'            ⊳ local search
15:       if x" met with all railway constraints then   ⊳ check all constraints
16:          solution (x") = calculate minimise a total delay (x")
17:       end if
18:       if solution (x") < solution (x) then          ⊳ move or not
19:          x = x"
20:          k = 1                                      ⊳ return to first neighbourhood structure
21:       else if
22:          k = k + 1                                  ⊳ move to next neighbourhood structure
23:       end if
24:   end for
25: end while
26:end procedure
```

Algorithm 5.3 presents the pseudocode for deterministic RSP BR-VNS Algorithm. Firstly, the process starts by selecting the set of neighbourhood structures (Pseudocode: line 2). Secondly, we use the biased randomised algorithm to assign the probability to reorder the train sequence to generate an initial solution (Pseudocode: line 3 to line 6). Then, we repeat the whole VNS process as shaking phase, local search phase and move or not phase until the stopping condition is met (Pseudocode: line 8 to line 25). We also need to make sure that that all railway operational constraints are met at each step (Pseudocode: line 11 to line 13 and line 15 to line 17).

5.4. Computational experiments

We need to evaluate the performance of the BR-VNS, which is the proposed solution method in this chapter by comparing the results from the previous chapter. The same set of data from Chapter 3 was used, consisting of two different cases which are the UK and Thailand Rail network. All constraints follow details in section 3.5.

The major aim of experimentation was to minimise the total train delays. BR-VNS was implemented on a personal computer using an Intel core i7-4500U CPU, 1.80GHz-2.4GHz, and 8GB RAM.

5.4.1. Southeastern train company, United Kingdom

The BR-VNS was compared to the result from the previous chapter and the basic VNS. In the real-life rescheduling process, we need to update the schedule within time limits, however CPLEX required more than 7 hours to reach the optimal solution. Therefore, we set 400 seconds computation time for all methods to compare between CPLEX, IG-BR, BR-IG-LS, VNS and BR-VNS to evaluate the efficiency of the BR-VNS.

Tables 5.1, 5.3, 5.5, 5.7, 5.9, 5.11 and 5.13 show the experimental results using CPLEX, IG-BR, BR-IG-LS, VNS and BR-VNS. In each table, all columns before the CPLEX column are used to provide detail of each instance. Moreover, the other columns show the total train delay of each instance by using a different solution method. Finally, at the bottom row of the table, we calculated an average of total delay.

Tables 5.2, 5.4, 5.6, 5.8, 5.10, 5.12 and 5.14 compare the average delay between each solution method from Tables 5.1, 5.3, 5.5, 5.7, 5.9, 5.11 and 5.13. Each sub column under the name of each solution shows Relative Deviation (RD) compared to all other solution methods proposed in this thesis as CPLEX (RD (%) compared with CPLEX), IG-BR (RD (%) compared with IG-BR), BR-IG-LS (RD (%) compared with BR-IG-LS) and VNS (RD (%) compared with VNS).

In addition, all delays are presented in **minutes** and a positive number of RD (%) means that the method has a better performance than the other methods, otherwise a negative number shows as no improvement.

Block Delay	# of Block(s)	# of Train(s)	Instance	CPLEX	IG-BR	BR-IG-LS	VNS	BR-VNS
				Delay	Delay	Delay	Delay	Delay
5	1	3	P01	107.83	97.67	95.33	95.33	**93.83**
	2	5	P02	143.67	**140.33**	150.50	145.50	143.67
10	1	3	P03	167.00	168.67	160.83	160.83	**156.50**
	2	5	P04	281.83	314.83	295.17	285.17	**280.83**
15	1	3	P05	260.17	247.33	247.33	252.17	**244.67**
	2	5	P06	314.67	313.33	**311.67**	**311.67**	**311.67**
20	1	9	P07	**1028.00**	1099.83	1073.50	1060.83	1061.83
	2	12	P08	**661.75**	705.50	731.00	704.25	702.25
25	1	9	P09	**1272.67**	1471.17	1308.83	1315.83	1288.00
	2	12	P10	1823.00	1829.00	**1816.67**	1834.83	1821.33
30	1	9	P11	**1021.00**	1023.33	1023.33	1023.33	1023.33
	2	12	P12	**1769.58**	2052.00	2051.83	1923.50	1894.33
Avg Delay				**737.60**	788.58	772.17	759.44	751.85

Table 5.1 : Deterministic Longer running/dwell times - minor and general disruption: BR-VNS

Block Delay	# of Block(s)	# of Train(s)	Instance	VNS			BR-VNS			
				RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with VNS
5	1	3	P01	11.59	2.40	0.00	12.99	3.93	1.57	1.57
	2	5	P02	-1.28	-3.68	3.32	0.00	-2.38	4.54	1.26
10	1	3	P03	3.69	4.65	0.00	6.29	7.22	2.69	2.69
	2	5	P04	-1.18	9.42	3.39	0.36	10.80	4.86	1.52
15	1	3	P05	3.07	-1.96	-1.96	5.96	1.08	1.08	2.97
	2	5	P06	0.95	0.53	0.00	0.95	0.53	0.00	0.00
20	1	9	P07	-3.19	3.55	1.18	-3.29	3.46	1.09	-0.09
	2	12	P08	-6.42	0.18	3.66	-6.12	0.46	3.93	0.28
25	1	9	P09	-3.39	10.56	-0.53	-1.20	12.45	1.59	2.12
	2	12	P10	-0.65	-0.32	-1.00	0.09	0.42	-0.26	0.74
30	1	9	P11	-0.23	0.00	0.00	-0.23	0.00	0.00	0.00
	2	12	P12	-8.70	6.26	6.25	-7.05	7.68	7.68	1.52
Avg RD(%)				-2.96	3.70	1.65	**-1.93**	**4.66**	**2.63**	**1.00**

Table 5.2 : Deterministic Longer running/dwell times - minor and general disruption: BR-VNS Results (RD)

Tables 5.1 and 5.2 show the result of using the BR-VNS to solve the deterministic RSP on minor and general delays for longer running time and dwell time. The average delay was used to evaluate the performance between the proposed methods. As expected, CPLEX outperformed the other algorithms in terms of average delay. However, the optimal solution could not be achieved due to the 400 seconds computation time limit. For average total train delays, the results from CPLEX were 2.96% better than VNS and 1.93% better than BR-VNS. However, if we only focus on our proposed algorithms, BR-VNS provided lowest average delays than the others with 1% lower than VNS, 2.63% lower than BR-IG-LS and 4.66% lower than IG-BR.

Late Departure	# of Train(s)	Instance	CPLEX	IG-BR	BR-IG-LS	VNS	BR-VNS
			Delay	Delay	Delay	Delay	Delay
5	1	R01	**61.83**	70.83	68.67	68.17	67.83
	3	R02	**59.83**	65.50	62.83	63.00	60.33
	5	R03	**73.17**	75.17	80.33	75.50	74.83
10	1	R04	**70.67**	78.50	75.00	72.17	71.00
	3	R05	**82.17**	83.50	82.83	82.67	82.17
	5	R06	**138.00**	143.83	143.17	140.67	140.83
15	1	R07	**67.66**	68.67	68.67	68.67	68.67
	3	R08	**119.33**	127.33	124.83	124.83	121.17
	5	R09	**167.83**	171.33	171.33	171.33	171.33
20	6	R10	**182.33**	184.00	183.67	183.67	**182.33**
	9	R11	**240.50**	250.67	246.83	244.33	242.67
	12	R12	**334.17**	347.67	347.67	347.67	347.67
25	6	R13	**230.67**	240.33	233.00	233.00	231.78
	9	R14	**275.67**	281.67	280.00	277.67	278.50
	12	R15	**376.67**	385.17	381.33	380.17	378.83
Avg Delay			**165.37**	171.61	170.01	168.90	168.00

Table 5.3 : Deterministic Late departures - minor and general disruption: BR-VNS Results

Late Departure	# of Train(s)	Instance	VNS			BR-VNS			
			RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with VNS
5	1	R01	-10.25	3.76	0.73	-9.70	4.24	1.22	0.50
	3	R02	-5.29	3.82	-0.27	-0.83	7.89	3.98	4.24
	5	R03	-3.19	-0.44	6.01	-2.27	0.45	6.85	0.89
10	1	R04	-2.13	8.06	3.77	-0.47	9.55	5.33	1.62
	3	R05	-0.61	0.99	0.19	0.00	1.59	0.80	0.60
	5	R06	-1.93	2.20	1.75	-2.05	2.09	1.63	-0.11
15	1	R07	-1.49	0.00	0.00	-1.49	0.00	0.00	0.00
	3	R08	-4.61	1.96	0.00	-1.54	4.84	2.93	2.93
	5	R09	-2.08	0.00	0.00	-2.08	0.00	0.00	0.00
20	6	R10	-0.73	0.18	0.00	0.00	0.91	0.73	0.73
	9	R11	-1.59	2.53	1.01	-0.90	3.19	1.69	0.68
	12	R12	-4.04	0.00	0.00	-4.04	0.00	0.00	0.00
25	6	R13	-1.01	3.05	0.00	-0.48	3.56	0.52	0.52
	9	R14	-0.73	1.42	0.83	-1.03	1.13	0.54	-0.30
	12	R15	-0.93	1.30	0.30	-0.57	1.65	0.66	0.35
Avg RD(%)			-2.14	1.58	0.65	**-1.59**	**2.11**	**1.19**	**0.54**

Table 5.4 : Deterministic Late departures - minor and general disruption: BR-VNS Results (RD)

Table 5.3 shows the results of using the BR-VNS to solve the deterministic RSP on minor and general delays for late departure times. Table 5.4 was generated for comparison between the proposed algorithms. CPLEX also outperformed the other algorithms in terms of average delays as we expected. The execution time limit of 400 seconds resulted in no optimal solution for CPLEX. For average total train delays, results of CPLEX were 2.14% lower than VNS and 1.59% lower than BR-VNS. Moreover, when we only compared between IG-BR, BR-IG-LS, VNS and BR-VNS the best average delays value was implemented by using the BR-VNS with 0.54% lower than VNS, 1.19% lower than BR-IG-LS and 2.11% lower than IG-BR.

Block Delay	# of Block(s)	Instance	CPLEX	IG-BR	BR-IG-LS	VNS	BR-VNS
			Delay	Delay	Delay	Delay	Delay
40	4	HP01	7706.00	7379.33	7227.47	7183.58	**7095.32**
	5	HP02	7432.00	7260.75	7118.82	7039.90	**6980.77**
	6	HP03	7619.50	7250.47	7335.30	**7250.47**	**7250.47**
50	4	HP04	9396.67	8836.87	8180.50	8056.70	**7925.68**
	5	HP05	9342.33	9150.20	8997.90	8824.50	**8682.29**
	6	HP06	8529.00	7548.50	7494.00	7401.92	**7347.02**
60	4	HP07	11574.83	10429.50	9987.60	9885.87	**9793.20**
	5	HP08	9309.58	8795.90	8871.20	8772.67	**8706.13**
	6	HP09	9570.67	8497.00	8497.00	**8497.00**	**8497.00**
70	4	HP10	13182.67	12187.30	11879.10	**11734.13**	11801.50
	5	HP11	10716.58	9656.60	9575.33	9530.01	**9452.31**
	6	HP12	13006.42	11795.70	11685.17	11558.75	**11521.67**
80	4	HP13	15491.00	14262.90	14151.60	**14008.20**	14086.15
	5	HP14	14858.67	13645.60	13320.50	13241.83	**13172.80**
	6	HP15	15646.83	14103.50	14052.80	13987.33	**13901.33**
Avg Delay			10892.18	10053.34	9891.62	9798.19	**9747.58**

Table 5.5 : Deterministic Longer running/dwell times - major disruption: BR-VNS Results

Block Delay	# of Block(s)	Instance	VNS			BR-VNS			
			RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with VNS
40	4	HP01	6.78	2.65	0.61	7.92	3.85	1.83	1.23
	5	HP02	5.28	3.04	1.11	6.07	3.86	1.94	0.84
	6	HP03	4.84	0.00	1.16	4.84	0.00	1.16	0.00
50	4	HP04	14.26	8.83	1.51	15.65	10.31	3.11	1.63
	5	HP05	5.54	3.56	1.93	7.07	5.11	3.51	1.61
	6	HP06	13.21	1.94	1.23	13.86	2.67	1.96	0.74
60	4	HP07	14.59	5.21	1.02	15.39	6.10	1.95	0.94
	5	HP08	5.77	0.26	1.11	6.48	1.02	1.86	0.76
	6	HP09	11.22	0.00	0.00	11.22	0.00	0.00	0.00
70	4	HP10	10.99	3.72	1.22	10.48	3.17	0.65	-0.57
	5	HP11	11.07	1.31	0.47	11.80	2.12	1.28	0.82
	6	HP12	11.13	2.01	1.08	11.42	2.32	1.40	0.32
80	4	HP13	9.57	1.79	1.01	9.07	1.24	0.46	-0.56
	5	HP14	10.88	2.96	0.59	11.35	3.46	1.11	0.52
	6	HP15	10.61	0.82	0.47	11.16	1.43	1.08	0.61
Avg RD(%)			10.04	2.54	0.94	**10.51**	**3.04**	**1.46**	**0.52**

Table 5.6 : Deterministic Longer running/dwell times - major disruption: BR-VNS Results (RD)

Tables 5.5 and 5.6 present the results of using the BR-VNS to solve the deterministic RSP on major delays for longer running time and dwell times. All implementations were based on 400 seconds for fair comparison and CPLEX was stopped before reaching the optimal. Results showed that the smallest average delay was computed by BR-VNS with 0.52% less than VNS, 1.46% less than BR-IG-LS, 3.04% less than IG-BR and 10.50% less than CPLEX. Therefore, the BR-VNS outperformed for solving the major disruptions of longer running time and dwell time.

Block Delay	# of Block(s)	Departure Delay	# of Train(s)	Instance	CPLEX	IG-BR	BR-IG-LS	VNS	BR-VNS
					Delay	Delay	Delay	Delay	Delay
40	4	10	3	PR01	7524.00	7379.92	7347.67	7298.36	**7208.22**
			5	PR02	7455.67	7285.42	7311.70	7267.67	**7135.00**
		20	9	PR03	7599.17	6975.83	6884.33	6789.83	**6697.69**
			12	PR04	7823.50	7107.00	7017.20	6977.33	**6853.13**
40	6	10	3	PR05	6830.00	6930.50	6816.00	**6732.75**	6788.67
			5	PR06	7773.56	7002.83	6997.75	6889.20	**6721.36**
		20	9	PR07	6518.50	6465.85	6515.83	6432.85	**6374.40**
			12	PR08	8088.50	7458.50	7333.67	7267.13	**7122.92**
50	4	10	3	PR09	9462.17	9346.13	9212.50	9123.27	**9041.33**
			5	PR10	9615.75	9097.40	8849.17	8754.69	**8667.22**
		20	9	PR11	9589.00	**7672.54**	**7672.54**	**7672.54**	**7672.54**
			12	PR12	9863.33	9194.50	9087.30	8997.40	**8805.45**
50	6	10	3	PR13	9346.42	8400.54	8367.80	8210.36	**8150.75**
			5	PR14	9737.17	**9741.20**	**9741.20**	**9741.20**	**9741.20**
		20	9	PR15	8609.00	8529.00	8432.40	8322.50	**8235.48**
			12	PR16	8286.92	8012.83	7967.08	7874.17	**7762.17**
Avg Delay					8382.67	7912.50	7847.13	7771.95	**7686.10**

Table 5.7 : Deterministic Longer running/dwell times and Late departures - major disruption: BR-VNS Results

Block Delay	# of Block(s)	Departure Delay	# of Train(s)	Instance	VNS			BR-VNS			
					RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with VNS
40	4	10	3	PR01	3.00	1.11	0.67	4.20	2.33	1.90	1.24
			5	PR02	2.52	0.24	0.60	4.30	2.06	2.42	1.83
		20	9	PR03	10.65	2.67	1.37	11.86	3.99	2.71	1.36
			12	PR04	10.82	1.82	0.57	12.40	3.57	2.34	1.78
40	6	10	3	PR05	1.42	2.85	1.22	0.61	2.05	0.40	-0.83
			5	PR06	11.38	1.62	1.55	13.54	4.02	3.95	2.44
		20	9	PR07	1.31	0.51	1.27	2.21	1.41	2.17	0.91
			12	PR08	10.15	2.57	0.91	11.94	4.50	2.87	1.98
50	4	10	3	PR09	3.58	2.38	0.97	4.45	3.26	1.86	0.90
			5	PR10	8.95	3.77	1.07	9.86	4.73	2.06	1.00
		20	9	PR11	19.99	0.00	0.00	19.99	0.00	0.00	0.00
			12	PR12	8.78	2.14	0.99	10.73	4.23	3.10	2.13
50	6	10	3	PR13	12.16	2.26	1.88	12.79	2.97	2.59	0.73
			5	PR14	-0.04	0.00	0.00	-0.04	0.00	0.00	0.00
		20	9	PR15	3.33	2.42	1.30	4.34	3.44	2.34	1.05
			12	PR16	4.98	1.73	1.17	6.33	3.13	2.57	1.42
Avg RD(%)					7.29	1.78	0.96	**8.31**	**2.86**	**2.05**	**1.10**

Table 5.8 : Deterministic Longer running/dwell times and Late departures - major disruption: BR-VNS Results (RD)

Table 5.7 highlights the results of using the BR-VNS to solve the deterministic RSP on major disruption on late departures time. Then, Table 5.8 provides a comparison between CPLEX, IG-BR, BR-IG-LS, VNS and BR-VNS. In this set of instances, BR-VNS gave high performance to reduce the average train delays at 1.10% lower than VNS, 2.05% lower than BR-IG-LS, 2.86% lower than IG-BR and 8.30% lower than CPLEX. Moreover, please note that the CPLEX solution was not optimal because of the 400 seconds time limit. Therefore, the BR-VNS provided a better value of average delays in less computation time.

As mentioned before, we separated the instance sets into 2 groups depending on the size of disruption as minor and general size of disruption and major size of disruption. Computation time to generate the experimental results was set as 400 seconds. This is in the time frame limit required to complete the rescheduling process and update the service as suggested by SSDP. Therefore, we summarised the results dependent on size of delay separately as follows:

Disruption Type		CPLEX	IG-BR	BR-IG-LS	VNS	BR-VNS
Minor and General	Longer Running/dwell times	**737.60**	788.58	772.17	759.44	751.85
	Late Departures	**165.37**	171.61	170.01	168.90	168.00
Average Delay		**451.48**	480.10	471.09	464.17	459.92

Table 5.9 : Summary of Experimental result of BR-VNS - Deterministic (UK) - Small

Disruption Type		VNS			BR-VNS			
		RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with VNS
Minor and General	Longer Running/dwell times	-2.96	3.70	1.65	-1.93	4.66	2.63	0.52
	Late Departures	-2.14	1.58	0.65	-1.59	2.11	1.19	1.10
Average Delay		-2.81	3.32	1.47	**-1.87**	**4.20**	**2.37**	**0.78**

Table 5.10 : Summary of Experimental result of BR-VNS - Deterministic (UK) - Small (RD)

Tables 5.9 and 5.10 show the summary results of small disruption instances. When we compared between CPLEX, IG-BR, BR-IG-LS, VNS and BR-VNS the best average train delay was provided by CPLEX at 2.81% better than VNS and 1.87% better than BR-VNS in the same computation time. This computation time was not long enough for CPLEX to achieve the optimal value; however, it can help to evaluate the performance of the proposed method. In addition, if we only focus on other proposed algorithms except CPLEX, the BR-VNS provided lowest average delays than the others with 0.78% lower than VNS, 2.37% lower than BR-IG-LS and 4.20% lower than IG-BR.

Disruption Type		CPLEX	IG-BR	BR-IG-LS	VNS	BR-VNS
Major	Longer Running/dwell times	10892.18	10053.34	9891.62	9798.19	**9747.58**
	Late Departures & Longer Running/dwell times	8382.67	7912.50	7847.13	7771.95	**7686.10**
Average Delay		9637.42	8982.92	8869.38	8785.07	**8716.84**

Table 5.11 : Summary of Experimental result of BR-VNS - Deterministic (UK) - Large

Disruption Type		VNS			BR-VNS			
		RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with VNS
Major	Longer Running/dwell times	10.04	2.54	0.94	10.51	3.04	1.46	0.52
	Late Departures & Longer Running/dwell times	7.29	1.78	0.96	8.31	2.86	2.05	1.10
Average Delay		8.84	2.20	0.95	**9.55**	**2.96**	**1.72**	**0.78**

Table 5.12 : Summary of Experimental result of BR-VNS - Deterministic (UK) - Large (RD)

Tables 5.11 and 5.12 show the summary results for large disruption instances. Results of CPLEX, IG-BR, BR-IG-LS, VNS and BR-VNS were compared with the proposed method. The BR-VNS had higher efficiency to reduce the total delays at 9.55% lower than CPLEX, 2.96% lower than IG-BR, 1.72% lower than BR-IG-LS and 0.78% lower than VNS. Note that CPLEX was terminated at 400 seconds before reaching the optimal solution.

Disruption Type		CPLEX	IG-BR	BR-IG-LS	VNS	BR-VNS
Minor and General	Longer Running/dwell times	**737.60**	788.58	772.17	759.44	751.85
	Late Departures	**165.37**	171.61	170.01	168.90	168.00
Major	Longer Running/dwell times	10892.18	10053.34	9891.62	9798.19	**9747.58**
	Late Departures & Longer Running/dwell times	8382.67	7912.50	7847.13	7771.95	**7686.10**
Average Delay		5044.45	4731.51	4670.23	4624.62	**4588.38**

Table 5.13 : Summary of Experimental result of BR-VNS - Deterministic (UK) - All

Disruption Type		VNS			BR-VNS			
		RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with VNS
Minor and General	Longer Running/dwell times	-2.96	3.70	1.65	-1.93	4.66	2.63	1.00
	Late Departures	-2.14	1.58	0.65	-1.59	2.11	1.19	0.54
Major	Longer Running/dwell times	10.04	2.54	0.94	10.51	3.04	1.46	0.52
	Late Departures & Longer Running/dwell times	7.29	1.78	0.96	8.31	2.86	2.05	1.10
Average Delay		8.32	2.26	0.98	**9.04**	**3.03**	**1.75**	**0.78**

Table 5.14 : Summary of Experimental result of BR-VNS - Deterministic (UK) - All (RD)

Tables 5.13 and 5.14 present a summary of all results in section 5.4.1.1. According to the literature review, CPLEX solution required long computation time to solve the complex RSP. Thus, stopping at 400 seconds did not achieve an optimal solution. However, CPLEX also proved effective for solving minor and general disruptions of RSP, while BR-VNS was the best choice for the major disruptions. The overview of this chapter concluded that BR-VNS improved the performance of deterministic RSP with the aim to minimise the total train delay by 0.78% better than VNS, 1.75% better than BR-IG-LS, 3.03% better than IG-BR and 9.04% better than CPLEX.

In addition, comparison between VNS and BR-VNS showed that biased randomisation improved the efficiency of basic VNS.

5.4.2. State Railway of Thailand

The computational experimental results compared historical data, CPLEX, IG-BR, BR-IG-LS, basic VNS and BR-VNS based on a 400 seconds time frame limit as suggested by SSDP to update the train service.

Table 5.15 shows the experimental results of historical data, CPLEX, IG-BR, BR-IG-LS, VNS and BR-VNS. In this table, the first two columns show the test instances, while the other columns show the total train delays of all methods studied in this thesis. The average of total delay is shown at the end of the table and all delays are shown in **minutes**.

Date	Instance	Historical Data	CPLEX	IG-BR	BR-IG-LS	VNS	BR-VNS
1/2/2019	THAF101	2477	2121	1973	1937	1882	**1866**
2/2/2019	THAF102	2138	2035	1998	1958	1905	**1838**
3/2/2019	THAF103	1830	1803	1789	1755	**1735**	**1735**
4/2/2019	THAF104	1681	1325	1154	1187	1108	**1061**
5/2/2019	THAF105	2549	2379	2247	2206	2167	**2048**
6/2/2019	THAF106	1750	1258	1234	1198	1136	**1125**
7/2/2019	THAF107	2329	2188	2107	2093	2081	**2062**
8/2/2019	THAF108	3186	2648	2422	2434	2399	**2349**
9/2/2019	THAF109	2143	1907	1895	1822	**1788**	1801
10/2/2019	THAF110	2180	1574	1535	1509	1464	**1411**
11/2/2019	THAF111	2712	1982	1825	1782	1741	**1713**
12/2/2019	THAF112	2472	1830	1794	1742	**1636**	1655
13/2/2019	THAF113	2440	1874	1713	1713	1683	**1647**
14/2/2019	THAF114	1807	1807	1780	1704	1654	**1600**
15/2/2019	THAF115	2903	2458	2265	2268	2202	**2156**
16/2/2019	THAF116	3582	3030	2877	2655	2580	**2440**
17/2/2019	THAF117	3516	2740	2554	2422	2361	**2248**
18/2/2019	THAF118	2895	2645	2321	2278	**2189**	2197
19/2/2019	THAF119	1880	1457	1398	1328	1299	**1254**
20/2/2019	THAF120	2791	1861	1801	1715	1765	**1687**
21/2/2019	THAF121	3798	3050	2821	2845	**2741**	2751
22/2/2019	THAF122	3658	3421	3293	3287	3178	**3119**
23/2/2019	THAF123	3772	3543	3355	3317	3108	**3215**
24/2/2019	THAF124	3597	3260	3180	3034	3057	**2978**
25/2/2019	THAF125	3974	3477	3321	3358	3278	**3211**
26/2/2019	THAF126	4873	4495	4288	4251	4204	**4152**
27/2/2019	THAF127	3385	3392	3275	3180	3107	**3015**
28/2/2019	THAF128	3023	2584	2447	2331	2277	**2220**
Average Delay		2833.61	2433.71	2309.36	2261.04	2204.46	**2162.64**

Table 5.15 : Deterministic Thai – BR-VNS Results

Table 5.16 compares the average delay between each solution method from Table 5.15. Each sub column under the name of each solution shows Relative Deviation (RD) with compared to all other solution methods proposed in this thesis as CPLEX (RD (%) compared with CPLEX), IG-BR (RD (%) compared with IG-BR), BR-IG-LS (RD (%) compared with BR-IG-LS) and VNS (RD (%) compared with VNS). In addition, a positive number of RD (%) means that the method showed an improvement when compared with the other methods.

Date	Instance	VNS				BR-VNS				
		RD (%) compared with Historical	RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with Historical	RD (%) compared with CPLEX	RD (%) compared with IG-BR	RD (%) compared with BR-IG-LS	RD (%) compared with VNS
1/2/2019	THAF101	24.02	11.27	4.61	2.84	24.67	12.02	5.42	3.67	0.85
2/2/2019	THAF102	10.90	6.39	4.65	2.71	14.03	9.68	8.01	6.13	3.52
3/2/2019	THAF103	5.19	3.77	3.02	1.14	5.19	3.77	3.02	1.14	0.00
4/2/2019	THAF104	34.09	16.38	3.99	6.66	36.88	19.92	8.06	10.61	4.24
5/2/2019	THAF105	14.99	8.91	3.56	1.77	19.65	13.91	8.86	7.16	5.49
6/2/2019	THAF106	35.09	9.70	7.94	5.18	35.71	10.57	8.83	6.09	0.97
7/2/2019	THAF107	10.65	4.89	1.23	0.57	11.46	5.76	2.14	1.48	0.91
8/2/2019	THAF108	24.70	9.40	0.95	1.44	26.27	11.29	3.01	3.49	2.08
9/2/2019	THAF109	16.57	6.24	5.65	1.87	15.96	5.56	4.96	1.15	-0.73
10/2/2019	THAF110	32.84	6.99	4.63	2.98	35.28	10.36	8.08	6.49	3.62
11/2/2019	THAF111	35.80	12.16	4.60	2.30	36.84	13.57	6.14	3.87	1.61
12/2/2019	THAF112	33.82	10.60	8.81	6.08	33.05	9.56	7.75	4.99	-1.16
13/2/2019	THAF113	31.02	10.19	1.75	1.75	32.50	12.11	3.85	3.85	2.14
14/2/2019	THAF114	8.47	8.47	7.08	2.93	11.46	11.46	10.11	6.10	3.26
15/2/2019	THAF115	24.15	10.41	2.78	2.91	25.73	12.29	4.81	4.94	2.09
16/2/2019	THAF116	27.97	14.85	10.32	2.82	31.88	19.47	15.19	8.10	5.43
17/2/2019	THAF117	32.85	13.83	7.56	2.52	36.06	17.96	11.98	7.18	4.79
18/2/2019	THAF118	24.39	17.24	5.69	3.91	24.11	16.94	5.34	3.56	-0.37
19/2/2019	THAF119	30.90	10.84	7.08	2.18	33.30	13.93	10.30	5.57	3.46
20/2/2019	THAF120	36.76	5.16	2.00	-2.92	39.56	9.35	6.33	1.63	4.42
21/2/2019	THAF121	27.83	10.13	2.84	3.66	27.57	9.80	2.48	3.30	-0.36
22/2/2019	THAF122	13.12	7.10	3.49	3.32	14.73	8.83	5.28	5.11	1.86
23/2/2019	THAF123	17.60	12.28	7.36	6.30	14.77	9.26	4.17	3.08	-3.44
24/2/2019	THAF124	15.01	6.23	3.87	-0.76	17.21	8.65	6.35	1.85	2.58
25/2/2019	THAF125	17.51	5.72	1.29	2.38	19.20	7.65	3.31	4.38	2.04
26/2/2019	THAF126	13.73	6.47	1.96	1.11	14.80	7.63	3.17	2.33	1.24
27/2/2019	THAF127	8.21	8.40	5.13	2.30	10.93	11.11	7.94	5.19	2.96
28/2/2019	THAF128	24.68	11.88	6.95	2.32	26.56	14.09	9.28	4.76	2.50
Average Delay		22.20	9.42	4.54	2.50	**23.68**	**11.14**	**6.35**	**4.35**	**1.90**

Table 5.16 : Deterministic Thai – BR-VNS Results (RD)Thai – BR-VNS Results (RD)

In Table 5.16 BR-VNS had the smallest average delay and outperformed all the other algorithms. By comparison, BR-VNS provided lower average train delays than the company's current solution by 23.68%. We compared our proposed method in this chapter using 400 seconds as the stopping condition for all experiments. The BR-VNS was outstanding among the other algorithms and decreased average total delays at 1.90% lower than VNS, 4.35% lower than BR-IG-LS, 6.35% lower than IG-BR and 11.14% lower than CPLEX. Moreover, please note that the solution from CPLEX was not optimal because we did not have time to wait until it reached that point (more than 15 hours). Therefore, we need to suddenly generate a new schedule to deal with the uncertainties.

In addition, as we expected, all proposed methods in this thesis can be used to improve the Thai rail management system. The company's current solution is based on FCFS using a manual system. We recommended that the company should use this optimisation technique to improve the management system.

5.5. Chapter Conclusion

The main aim of this chapter was to use another metaheuristic method that improved the solution quality and decreased the gap of results from the previous chapter. BR-VNS was developed to satisfy the aim of this chapter with the major contribution of improving the quality of the solution. In more detail, VNS is commonly and successfully used for implementation of COPs. From previous chapters, biased randomisation combined with IG and IG-LS provided a good improvement which is the reason why we selected the BR-VNS to solve the deterministic RSP. In this chapter, we also used the optimisation model with the objective to minimise the total train delays.

The procedure of BR-VNS starts by using biased randomisation to generate the initial solution, then the basic concept of VNS to improve the solution quality. The basic VNS consists of two or more neighbourhood structures which usually involve insertion or swap search. The VNS algorithm process is (i) shake phase to generate new random initial values to avoid cycling, (ii) local search phase by using swap moves or insert moves to find a local optimal and (iii) move or not phase for acceptance criterion. If the solution is improved, the best solution is updated with continuous search in the first neighbourhood structure. Alternatively, if the new solution is not improved, the algorithm moves to the next neighbourhood structure.

We compared the proposed method with the results of the real-world case studies from Southeastern train company, UK (Khosravi, 2013) and State Railway of Thailand from the previous chapter. All experiments were run within the 400 seconds time frame. In the UK test instances, the overall results showed that BR-VNS outperformed all the other algorithms for the deterministic RSP; however, on the minor and general disruption set of instances, CPLEX provided a small gap better than BR-VNS. For the Thailand case study, experimental results showed that the BR-VNS had high performance compared to the company’s historical solution, IG-BR, and BR-IG-LS. Therefore, the BR-VNS was more effective in solving large and complex problems within the limited execution time.

The experimental results in this chapter clearly showed that the BR-VNS algorithm was the best choice to solve the deterministic RSP. The company can decrease delay propagation, increase the system efficiency and increase benefits for customer satisfaction.

Part II

Stochastic Railway Scheduling in the presence of uncertainties

Chapter 6: Sim-Iterated Greedy with Biased Randomised (S-IG-BR) for stochastic RSP

6.1. Introduction

In the previous chapter, we proposed to solve RSP by considering deterministic delay time. In this chapter, we set a model to consider the random value of delay. This type of problem can be called stochastic RSP and aims to minimise the total delay of all trains in the network. The overview of stochastic RSP is described and added as stochastic behaviour to an optimisation model for simulating a real-world complex situation. In the optimisation model, all characteristics of the railway network are defined in terms of objective function, decision variable and constraints which mainly focus on headway and signalling constraints. Moreover, for the stochastic situation, we generate random delay which is added into the running/dwell time and departure time constraints. The solution methods consist of simulation to generate the stochastic behaviour into the model and heuristics or metaheuristics called sim-optimisation. We address this problem because it can improve the reliability and efficiency of the algorithms. To the best of our knowledge, no previous paper has considered sim-optimisation to solve the stochastic RSP. Therefore, this is an opportunity to implement sim-optimisation to improve the deterministic RSP solution methods.

Results from Chapter 3 showed that IG-BR has good performance to solve the deterministic RSP; therefore, we propose the sim-optimisation method which combines simulation techniques as Monte Carlo Simulation (MCS) and IG-BR to solve the stochastic RSP. The MCS technique is used to model the probability distribution for a process that is unpredictable. We generated a random value of delay before we started to solve the problem and these values should be added into the main optimisation model. We found only one paper that used MCS to solve a topic related to the RSP in the literature. Ushida et al. (2011) focused on using MCS to increase robustness of the timetables. They used a Chromatic Diagram to collect characteristics of delay at the station for adding a buffer time. This is an extra period of time that is excluded from the dwell time or running time to generate a robust timetable. The MCS was used to random the dwell time delay value for testing the realistic of the problem. The result showed that the algorithm worked with real-world timetable data from the Japan rail network. Another topic, very similar to RSP was studied by Juan et al. (2014a). They used MCS with ILS heuristic to solve the permutation flow shop problem. In this problem, the value of the processing time was uncertain, therefore they used MCS to randomly generate processing time delay before solving a problem. Results showed that this algorithm was effective for the stochastic RSP. Therefore, in this chapter, we proposed MCS, which is the technique that successful with other COPs, combined with the IG-BR algorithm to solve the RSP that deals with a stochastic delay.

This chapter discusses the basic concept of simulation optimisation, and MCS. It shows how to apply the MCS to the IG-BR algorithm for solving a stochastic RSP. The combination of this proposed solution method is called Sim-Iterated Greedy with Biased Randomised (S-IG-BR). We also provide an explanation of how to compare between the deterministic and stochastic results in both UK and Thailand rail network test instances to measure the efficiency of the proposed methods.

The chapter is structured as follows: section 6.2 presents this chapter's contributions, then the stochastic RSP optimisation model is proposed in section 6.3. Section 6.4 presents the proposed S-IG-BR algorithm to solve RSP with stochastic disruption delays. All the experimental results are shown in section 6.5 which is separated into two sets of case studies (UK & Thailand). Section 6.6 is a conclusion of this chapter.

6.2. Contributions

Most of the literature on RSP focused on deterministic delays that are known in advance. However, in the real-world situation, we do not know the delay value before it occurs; therefore, the MCS simulation technique is commonly used to simulate the real-life environment for COPs. Juan et al. (2014) claimed that simulation optimisation with heuristics and/or metaheuristics generated a good solution with consideration on random variable values for the stochastic COPs.

The main contribution of this chapter is firstly, to develop a stochastic RSP optimisation model which considers random delay. Secondly, to develop and implement a hybrid algorithm which combines IG-BR and MCS techniques, called S-IG-BR, for solving the stochastic RSP. The IG-BR is used to find the solution and aims to minimise the total delay of all trains, while MCS generates random delays to simulate a real environment of the case study. In addition, MCS can randomly generate delays for the stochastic RSP.

To the best of our knowledge, the IG-BR with MCS has not been used to solve the stochastic RSP before. Thus, we are the first to present the S-IG-BR algorithm for solve the RSP with stochastic delay.

6.3. Stochastic RSP optimisation model

For the stochastics RSP optimisation model, we used an MCS to simulate the real situation environment for the disruption which occurs randomly in the system by using mean and variance (generate random delay into the system). Moreover, in this thesis, we only focused on late departure, dwell and travelling delay. However, the objective function of stochastic RSP is similar to deterministic RSP, which is to minimise total weight of train delays ($equation$ 5.1).

$$Minimise \quad z = \sum_{i \in I} w_i T_i \tag{6.1}$$

The set of constraints in the stochastic RSP optimisation model is the same as that used in the deterministic RSP optimisation model which presented in section 3.3.2., however random values which generated by MCS were added in the original optimisation model as follows:

We added the random values of departure time delay (rv_i) into departure times (r_i) to satisfy the late departure delay ($equation$ 6.2):

$$r_i = r_i + rv_i \qquad i \in I \tag{6.2}$$

and other random values of running or dwell time delay ($pv_{i,m_{i,k}}$) were added into running times ($p_{i,m_{i,k}}$) to satisfy running and dwell time delay ($equation$ 6.3):

$$p_{i,m_{i,k}} = p_{i,m_{i,k}} + pv_{i,m_{i,k}} \qquad i \in I; \quad k = 2, 3, \dots, l_i \tag{6.3}$$

6.4. Proposed Sim-IG with Biased Randomised (S-IG-BR) for solving RSP

Simulation-based optimisation (Sim-optimisation) was introduced by Glover et al. (1996) to measure the efficiency and reliability of optimisation methods for a real-world complex case study. Sim-optimisation combines stochastic nature and optimisation nature. Stochastic nature operates using a simulation model with two steps of (i) choose a specific variable and (ii) define a probability distribution to generate a random value into the optimisation model. For the optimisation nature, we used a heuristics or metaheuristics approach to solve the optimisation model, find a near-optimal solution and then repeated the whole process until the stopping condition was met. This method provided *n* number of solutions to find an average objective function value (Glover et al., 1996; Glover et al., 1999). Moreover, the sim-optimisation method research in COPs has been widely studied after introduction by Glover et al. (1996), mostly applied to manufacturing problems.

Chong et al. (2003) introduced discrete-event simulation to solve real-time scheduling mechanism manufacturing. This paper aimed to undertake rescheduling under unpredictable production disturbances for example machine error. Klemmt et al. (2009) also used discrete-event simulation based on the mixed integer programming (MIP) to generate a scheduling problem. Frantzén et al. (2011) studied real-time optimisation and decision-making support by using a sim-optimisation based on a GA to re-generate feasible schedules when disturbances occurred.

Arakawa et al. (2003) set a model to eliminate tardiness from a job shop production schedule with the capacity adjustment function using an optimisation-oriented simulation-based method. Moreover, sim-optimisation based on a GA was used to solve a flexible job shop scheduling problem by Nicoară et al. (2011). They set a model to find multi-objectives which minimised a total makespan, number of late jobs and average idle times of the job in process. Furthermore, Korytkowski et al. (2013) used an evolutionary simulation-based heuristic to solve a dynamic job shop scheduling. The conditions of the problem which they considered were the dispatching rule and the sum of internal and external delays. Moreover, Yang et al. (2004) developed a discrete-event simulation under a selected condition to model a stochastic problem for flow shop with multiple processors based on TS algorithm. This provided an effective schedule.

6.4.1. Monte Carlo Simulation (MCS)

We proposed a sim-heuristic algorithm which combined simulation with a heuristic for optimising our algorithm performance. In this thesis, the MCS was used as probability distribution for a generating the expected delays which are the unpredictable events by using mean and variance of the historical data or test instances, like a real situation where we do not know when the disruption will occur. Therefore, we used MCS to randomly generate the expected delays to represent a stochastic problem.

Algorithm 6.1 shows the process of MCS for solving the problems; it starts by generating the initial solution and makes use of MCS to estimate the expected random values of running time and departure time delay (Pseudocode: line 8). After that, input this solution to run using some heuristic to find the solution of each iteration. In this case, the heuristic that we use is IG with Biased Randomised (Pseudocode: line 9). Therefore, sum all solutions together and find the average (Pseudocode: line 10 to line 14).

Algorithm 6.1 : Procedure for MCS

1: **Procedure for MCS** (r',i , α)

▻ r': Initial solution

▻ i: Number of iterations

▻ α: Random value of delay

2: heuristic <- DefineHeuristic(input)

3: r': *GenerateInitialSolution*(heuristic, input); ▻ *Initial solution*

4: probaDist <- DefindProbabilityDistribution(input)

5: iteration = 0;

6: **while** stopping criteria are not satisfied **do** ▻ *loop until its reaches stopping condition*

7: **for** each iteration *i* **do** ▻ *start Monte Carlo simulation*

8: GenerateRandomDelay(heuristic, probaDist, input) ▻ *randomly generate delay*

9: solution (r^{ans}) = Heuristic(input, α)

10: solution (r^{best}) = solution (r^{best}) + solution (r^{ans}) ▻ *Sum up all solution (r^{ans})*

11: iteration++ ▻ *Count number of iterations*

12: **end for**

13: **end while**

14: averageDelay = solution (r^{best}) / iteration ▻ *Average the total delay*

15:**end procedure**

6.4.2. Sim-IG with Biased Randomised (S-IG-BR)

To solve stochastic COPs is very challenging but necessary. Thus, it is most important to use the best simulation techniques to generate scenarios close to real-world situations. Results of IG-BR on the deterministic situation in Chapter 3 provided a good solution for the deterministic RSP. Therefore, we considered using the S-IG-BR to solve the stochastic RSP. Hence, this chapter proposed an extension to the IG-BR algorithm by applying MCS in IG-BR algorithm in order to improve the solution of the stochastic RSP. The main procedure is as follows: (a) Initial solution based on given timetables in our case studies which comprises of real-world data from the UK and Thailand Rail network, (b) MCS process which randomly generates expected delays, (c) IG is used to solve the RSP by using two main phases (destruction and construction) and (d) the use of biased randomised algorithm to improve the resulting solution (Algorithm 6.2).

Algorithm 6.2 : Procedure for Stochastic RSP S-IG-BR Algorithm

---

1: **Procedure for Stochastic RSP S-IG-BR Algorithm** (r', r^D, r^R, d, b, β)
⊳ r': Initial solution
⊳ r^D: Partial sequence to reconstruct
⊳ r^R: Trains to reinsert
⊳ d: Random chosen number of trains
⊳ b: Number of blocks
⊳ β: Parameter for biased randomised
2: r': *GenerateInitialSolution*(); ⊳ *Initial solution*
3: iteration = 0;
4: **while** execution_time < 400 **do** ⊳ *Execution time 400 seconds*
5: **for** each train in b **do** ⊳ *start Monte Carlo simulation*
6: trainDuration = trainDuration + generateRandomDelay ⊳ *randomly generate delay*
7: **end for**
8: **for** i **=** 1 to b **do** ⊳ *run for all block*
9: set r^R = empty
10: **for** i **=** 1 to d **do** ⊳ *Destruction step*
11: r^R <- remove one node at randomly from r' and insert it in r^R;
12: $r^D = r'$ <- the remaining set of trains
13: **end for**
14: re-ordering r^R by using β to assign a probability ⊳ *Biased randomised*
15: **for** j = 1 to d **do** ⊳ *Construction step*
16: r^{ans} = best solution obtained after insert train from r^R in all possible position of r^D
17: **if** r^{ans} met with all railway constraints **then** ⊳ *check all constraints*
18: solution (r^{ans}) = calculate minimise a total delay (r^{ans})
19: **end if**
20: **end for**
21: **end for**
22 solution (r^{best}) = solution (r^{best}) + solution (r^{ans}) ⊳ *Sum up all solution (r^{ans})*
23: iteration++ ⊳ *Count number of iterations*
24: **end while**
25: averageDelay = solution (r^{best}) / iteration ⊳ *Average the total delay*
26:**end procedure**

Algorithm 6.2 presents the pseudocode for stochastic RSP S-IG-BR Algorithm. Firstly, the process starts by creating an initial solution from the rail timetable (Pseudocode: line 2). We consider the stochastic disruption situation, so the amount of the delay was generated by using MCS (Pseudocode: lines 5 to 7). The "generateRandomDelay" is an MCS function. This is shown in detail as an MCS process in Algorithm 6.1 (Pseudocode: lines 6).

Then, we start to implement the destruction phase of IG to find a new feasible solution (Pseudocode: line 10 to line 13). At the end of destruction phase, we need to apply the biased randomised algorithm to reorder the construction set by using a geometric probability method, before continuing to the next step (Pseudocode: line 14).

Then, in the construction phase, we reinsert the construction set back into all possible positions of the remaining set and save a position solution which provides the best solution (Pseudocode: line 15 to line 20). Moreover, in this step, we also need to confirm that all railway constraints are met (Pseudocode: line 17 to line 19).

Following the destruction and construction phases, the MCS process is used to find an expected average of the total delays (Pseudocode: line 22 to line 25).

6.5. Computational experiments

In this chapter, we proposed to evaluate the performance of the S-IG-BR by comparing it to the deterministic results. We executed all experiments on the same personal computer with Intel core i7-4500U CPU, 1.80GHz-2.4GHz, and 8GB RAM. Moreover, we used the same test instances as described in section 3.5.1.1. and 3.5.2.1.

In the stochastic RSP, we cannot directly compare between the deterministic solution and stochastic solution, therefore we added the stochastic scenario into the best deterministic solution by using MCS to generate the expected total train delays and reliability.

The process of generating the best deterministic solution in a stochastic scenario (BDS-Stoch) starts by setting the best deterministic solution in a deterministic scenario (BDS-Det) as the best solution that minimises total train delays obtained in the deterministic RSP from Chapter 6 as an initial solution. Then, we use the MCS process to generate the expected total train delays of the same solution with BDS-Det and add the stochastic scenario into the best deterministic solution called BDS-Stoch. This process also provides reliability to estimate the probability that the best deterministic solution can be successfully used in a stochastic scenario. After that, we compare the BDS-Stoch and our stochastic solution to measure the performance of the sim-optimisation methods.

In addition, the reliability of a solution was computed as follows:

$$R = \left(1 - \frac{\sum_{n=1}^{nSim} sequenceFailures}{nSim}\right) \tag{6.4}$$

From $equation$ 6.4, the reliability (R) should be calculated by 1 minus number error sequence which cannot found the solution ($sequenceFailures$) divided by number of iterations ($nSim$). Moreover, we transformed the level of reliability into percentages by R multiple by 100 for a clearer view of the comparison.

In addition, the computation time limit was also set at 400 seconds. Therefore, the solution of S-IG-BR was compared to BDS-Stoch to evaluate the performance and reliability of the solution method.

6.5.1. Southeastern train company, United Kingdom

The computational experiments were compared between BDS-Stoch, which is generated by using MCS to transform the best deterministic solution found in the previous chapter and S-IG-BR which is discussed in this chapter.

Tables 6.1-6.4 show the experimental results using BDS-Stoch and S-IG-BR to solve the UK case study in the stochastic situation. In each table, all columns before the column BDS-Stoch show the detail of each instance and all delays are shown in **minutes**. For column BDS-Stoch, we provide total train delay and reliability. Then, the columns S-IG-BR presents the results of total train delay, the reliability of the solution methods and the percentages of improvement between S-IG-BR and BDS-Stoch by using RD (%). Finally, at the bottom row of each table we calculated an average of the total delay, reliability and RD, which can provide a clearer comparison between the solution methods.

A positive number of RD (%) means that the result of our proposed method can provide solution better than original results. Otherwise, a negative result means that there are no improvements.

Block Delay	# of Block(s)	# of Train(s)	Instance	BDS-Stoch		S-IG-BR		RD (%) of S-IG-BR compared with BDS-Stoch
				Delay	Reliability	Delay	Reliability	
5	1	3	P01	124.66	0.56	119.10	0.96	4.46
	2	5	P02	159.88	0.72	155.35	0.98	2.84
10	1	3	P03	176.79	0.78	173.78	0.97	1.70
	2	5	P04	285.86	0.74	284.46	0.97	0.49
15	1	3	P05	266.81	0.57	266.48	0.96	0.12
	2	5	P06	316.62	0.84	316.45	1.00	0.05
20	1	9	P07	1133.35	0.78	1129.02	0.99	0.38
	2	12	P08	771.00	0.59	733.03	0.87	4.92
25	1	9	P09	1409.90	0.78	1408.45	0.92	0.10
	2	12	P10	1871.83	0.59	1871.06	0.88	0.04
30	1	9	P11	1108.30	0.77	1092.59	0.87	1.42
	2	12	P12	1950.01	0.73	1903.46	0.87	2.39
Avg Delay				797.92	0.70	**787.77**	**0.94**	**1.27**

Table 6.1 : Stochastic Longer running/dwell times - minor and general disruption: S-IG-BR Results

Table 6.1 shows the S-IG-BR results of stochastic RSP considered on minor and general delays on longer running time and dwell time. The best solution for each instance is indicated in **bold**. The comparison between BDS-Stoch and S-IG-BR showed that S-IG-BR reduced the expected total train delay at 1.27% better than the BDS-Stoch. Moreover, the reliability of S-IG-BR was higher than BDS-Stoch by 24%.

Late Departure	# of Train(s)	Instance	BDS-Stoch		S-IG-BR		RD (%) of S-IG-BR compared with BDS-Stoch
			Delay	Reliability	Delay	Reliability	
5	1	R01	67.41	0.52	67.14	0.99	0.40
	3	R02	67.89	0.77	66.37	0.99	2.23
	5	R03	82.51	0.81	76.83	0.98	6.88
10	1	R04	78.32	0.64	75.07	0.99	4.15
	3	R05	94.25	0.59	91.54	0.99	2.89
	5	R06	157.85	0.73	148.17	0.98	6.14
15	1	R07	76.75	0.83	70.46	0.99	8.20
	3	R08	132.96	0.88	123.80	0.98	6.89
	5	R09	190.61	0.79	184.97	0.98	2.96
20	6	R10	208.98	0.64	190.39	0.98	8.90
	9	R11	272.86	0.78	256.13	0.97	6.13
	12	R12	383.79	0.62	354.18	0.95	7.71
25	6	R13	261.57	0.66	260.62	0.98	0.36
	9	R14	315.59	0.60	305.60	0.96	3.17
	12	R15	431.69	0.74	415.41	0.93	3.77
Avg Delay			188.20	0.71	**179.11**	**0.98**	**4.83**

Table 6.2 : Stochastic Late departures - minor and general disruption: S-IG-BR Results

Table 6.2 shows the S-IG-BR results of stochastic RSP considered on minor and general delays of late departure times. The best solution for each instance is indicated in **bold**. The expected total train delay results which provided by S-IG-BR was 4.83% lower than the solution from BDS-Stoch. In terms of reliability, S-IG-BR had lower sequence failure than BDS-Stoch by 27%.

Block Delay	# of Block(s)	Instance	BDS-Stoch		S-IG-BR		RD (%) of S-IG-BR compared with BDS-Stoch
			Delay	Reliability	Delay	Reliability	
40	4	HP01	8112.99	0.33	7949.56	0.62	2.01
	5	HP02	8108.44	0.23	7837.70	0.98	3.34
	6	HP03	7374.15	0.57	7352.63	0.48	0.29
50	4	HP04	9011.98	0.45	8760.50	0.54	2.79
	5	HP05	9539.04	0.42	9239.39	0.71	3.14
	6	HP06	7436.10	0.31	7424.27	0.96	0.16
60	4	HP07	10356.47	0.49	10280.29	0.48	0.74
	5	HP08	9120.18	0.20	8905.48	0.57	2.35
	6	HP09	9551.90	0.52	9306.72	0.75	2.57
70	4	HP10	14344.90	0.35	13908.34	0.80	3.04
	5	HP11	11010.32	0.39	10795.17	0.74	1.95
	6	HP12	13558.67	0.22	13384.99	0.72	1.28
80	4	HP13	15505.31	0.60	15421.20	0.89	0.54
	5	HP14	14868.93	0.35	14690.12	0.75	1.20
	6	HP15	16221.56	0.50	16110.78	0.69	0.68
Avg Delay			10941.40	0.40	**10757.81**	**0.71**	**1.68**

Table 6.3 : Stochastic Longer running/dwell times - major disruption: S-IG-BR Results

Table 6.3 presents the S-IG-BR results of stochastic RSP considered for major delays of longer running time and dwell times. The best solution for each instance is indicated in **bold**. Results of this set of instances showed that S-IG-BR reduced the expected total train delays at 1.68% better than BDS-Stoch. Moreover, the reliability of S-IG-BR was 71% and BDS-Stoch was 40% with the gap of reliability at 31%.

Block Delay	# of Block(s)	Departure Delay	# of Train(s)	Instance	BDS-Stoch		S-IG-BR		RD (%) of S-IG-BR compared with BDS-Stoch
					Delay	Reliability	Delay	Reliability	
40	4	10	3	PR01	9306.78	0.45	8236.26	0.75	11.50
			5	PR02	8879.46	0.26	8652.64	0.52	2.55
		20	9	PR03	7923.36	0.56	7643.25	0.60	3.54
			12	PR04	8096.74	0.52	7905.48	0.59	2.36
40	6	10	3	PR05	7225.57	0.41	7180.96	0.71	0.62
			5	PR06	8720.84	0.33	8717.83	0.45	0.03
		20	9	PR07	6667.70	0.49	6666.19	0.98	0.02
			12	PR08	7755.11	0.50	7735.50	0.96	0.25
50	4	10	3	PR09	11441.82	0.18	10322.71	0.42	9.78
			5	PR10	10744.84	0.54	10406.84	0.55	3.15
		20	9	PR11	7884.00	0.31	7814.99	0.52	0.88
			12	PR12	10518.84	0.35	10415.75	0.60	0.98
50	6	10	3	PR13	8605.79	0.22	8532.53	0.66	0.85
			5	PR14	9827.53	0.56	9819.17	0.80	0.09
		20	9	PR15	8829.11	0.21	8701.99	0.57	1.44
			12	PR16	9693.25	0.21	9141.21	0.67	5.70
Avg Delay					8882.55	0.38	**8618.33**	**0.65**	**2.97**

Table 6.4 : Stochastic Longer running/dwell times and Late departures - major disruption: S-IG-BR Results

Table 6.4 indicates the S-IG-BR results of stochastic RSP considered on major disruption of late departure times. The best solution for each instance is indicated in **bold**. The overall expected average of the total train delay of S-IG-BR was lower than BDS-Stoch by 2.97%. Moreover, the reliability of S-IG-BR was 65% and 27% higher than BDS-Stoch.

The SSDP separated disruptions into 3 types depending on the size as minor disruptions, general disruptions and major disruptions. Therefore, we summarised and compared the results dependent on size of delay separately.

Disruption Type		BDS-Stoch		S-IG-BR		RD (%) compared with BDS-Stoch
		Delay	Reliability	Delay	Reliability	
Minor and General	Longer Running/dwell times	797.92	0.70	787.77	0.94	1.27
	Late Departures	188.20	0.71	179.11	0.98	4.83
Average Delay		493.06	0.71	**483.44**	**0.96**	**1.95**

Table 6.5 : Summary of Experimental result of S-IG-BR - Stochastic (UK) - Small

Disruption Type		BDS-Stoch		S-IG-BR		RD (%) compared with BDS-Stoch
		Delay	Reliability	Delay	Reliability	
Major	Longer Running/dwell times	10941.40	0.40	10757.81	0.71	1.68
	Late Departures & Longer Running/dwell times	8882.55	0.38	8618.33	0.65	2.97
Average Delay		9911.97	0.39	**9688.07**	**0.68**	**2.26**

Table 6.6 : Summary of Experimental result of S-IG-BR - Stochastic (UK) - Large

Tables 6.5 and 6.6 show the summary of results from Table. 6.1 to 6.4 considered separately depending on size of disruption. Table 6.5 presents the result of minor and general disruption. S-IG-BR decreased average train delay at 1.95% lower than BDS-Stoch. Results of major disruption are shown in Table 6.6. S-IG-BR reduced expected train delay by 2.26% when compared to BDS-Stoch.

The reliability of S-IG-BR decreased from 96% for minor and general disruption to 68% for major disruption. Reliability of BDS-Stoch also decreased from 71% in minor and general disruptions to 39% in major disruptions. When the level of uncertainty increased, the reliability decreased. Moreover, when we compared the reliability between S-IG-BR and BDS-Stoch, the result showed that S-IG-BR was more reliable at 25% higher than BDS-Stoch for minor and general delays and 29% more reliable than BDS-Stoch for major delays.

Disruption Type		BDS-Stoch		S-IG-BR		RD (%) compared with BDS-Stoch
		Delay	Reliability	Delay	Reliability	
Minor and General	Longer Running/dwell times	797.92	0.70	787.77	0.94	1.27
	Late Departures	188.20	0.71	179.11	0.98	4.83
Major	Longer Running/dwell times	10941.40	0.40	10757.81	0.71	1.68
	Late Departures & Longer Running/dwell times	8882.55	0.38	8618.33	0.65	2.97
Average Delay		5202.52	0.55	**5085.76**	**0.82**	**2.24**

Table 6.7 : Summary of Experimental result of S-IG-BR - Stochastic (UK) - All

Table 6.7 presents the overall picture of all results in section 6.5.1. S-IG-BR improved the quality of solution by 2.24% compared with the solution of BDS-Stoch. Moreover, S-IG-BR also provided 27% higher reliability than BDS-Stoch. In addition, the results also show that S-IG-BR had more efficiency than the deterministic methods for dealing with the uncertainties of delays in term of average results and reliability.

It should be noted that BDS-Stoch only used the MCS to generate the expected total train delays but did not consider the expected uncertainty values. Therefore, we used MCS to generate the expected value because this can be improved the performance and reliability of the methods.

6.5.2. State Railway of Thailand

The computational experiments were compared with BDS-Stoch which was generated by using MCS to transform the best deterministic solution found in the previous chapter and S-IG-BR which is discussed in this chapter.

Table 6.8 shows the experimental results of S-IG-BR and BDS-Stoch for solving the stochastic RSP of Thai rail network. In this table, the first two columns show the detail of each instance. For BDS-Stoch and S-IG-BR columns, we give the results of total delay and the reliability of the solution methods. Moreover, the last column shows the comparison of the total delays between S-IG-BR and BDS-Stoch. Finally, in the bottom row of the table, we calculated an average of the total delays, reliability and RD which can provide a clearer understanding. In addition, all delays time are presented in **minutes**. A positive number of RD shows that the proposed method provides an improvement.

Date	Instance	BDS-Stoch		S-IG-BR		RD (%) of S-IG-BR compared with BDS-Stoch
		Delay	Reliability	Delay	Reliability	
1/2/2019	THAF101	2140.26	0.46	2132.87	0.63	0.35
2/2/2019	THAF102	2222.09	0.43	2141.47	0.77	3.63
3/2/2019	THAF103	2069.37	0.33	2046.06	0.91	1.13
4/2/2019	THAF104	1547.01	0.26	1368.37	0.81	11.55
5/2/2019	THAF105	2289.36	0.24	2284.32	0.72	0.22
6/2/2019	THAF106	1665.72	0.52	1618.95	0.69	2.81
7/2/2019	THAF107	2637.56	0.20	2500.98	0.88	5.18
8/2/2019	THAF108	2768.12	0.68	2692.45	0.71	2.73
9/2/2019	THAF109	2571.86	0.67	1984.98	0.61	22.82
10/2/2019	THAF110	1643.44	0.21	1520.22	0.74	7.50
11/2/2019	THAF111	1855.58	0.70	1826.68	0.67	1.56
12/2/2019	THAF112	1981.48	0.53	1951.20	0.61	1.53
13/2/2019	THAF113	2101.69	0.32	1807.63	0.66	13.99
14/2/2019	THAF114	2117.11	0.33	1944.44	0.73	8.16
15/2/2019	THAF115	2866.43	0.53	2645.20	0.68	7.72
16/2/2019	THAF116	3026.04	0.55	2797.11	0.81	7.57
17/2/2019	THAF117	2285.48	0.65	2273.10	0.93	0.54
18/2/2019	THAF118	2583.71	0.26	2570.77	0.62	0.50
19/2/2019	THAF119	1689.85	0.65	1537.25	0.75	9.03
20/2/2019	THAF120	1854.07	0.60	1829.25	0.79	1.34
21/2/2019	THAF121	3916.94	0.44	3212.71	0.68	17.98
22/2/2019	THAF122	3993.31	0.51	3794.86	0.77	4.97
23/2/2019	THAF123	3767.96	0.27	3460.03	0.65	8.17
24/2/2019	THAF124	3055.96	0.48	3051.14	0.87	0.16
25/2/2019	THAF125	3561.76	0.22	3400.97	0.78	4.51
26/2/2019	THAF126	5145.14	0.52	5091.63	0.71	1.04
27/2/2019	THAF127	3595.63	0.36	3283.33	0.90	8.69
28/2/2019	THAF128	3207.96	0.53	2969.39	0.89	7.44
Average Delay		2648.60	0.44	**2490.62**	**0.75**	**5.96**

Table 6.8 : Stochastic Thai: S-BR-IG Results

Table 6.8 shows the results of stochastic RSP on the Thailand case study. The best solution for each instance is indicated in **bold**. In terms of the average expected total train delays, the solution of S-IG-BR was lower than BDS-Stoch by 5.96%. This means that sim-optimisation improved the solution of the deterministic RSP. For the reliability, the result shows that S-IG-BR was more reliable than the deterministic solution (BDS-Stoch) by 31%, because it was more suitable for solving the real-life situation of the uncertainties that randomly occur in the system. Moreover, the benefit of generating the expected running time, dwell time and departure time delays reduced expected total train delays and increased reliability.

6.6. Chapter Conclusion

This chapter examined the real-world problem of unexpected situations such as late departure, running time and dwell time delay which affect RSP. The stochastic optimisation model of RSP was developed and integrated with a solution methodology for dealing with the delays to increase solution flexibility for RSP. The optimisation model in this chapter was developed to minimise total train delays under uncertainty by considering the rescheduling process. The proposed combination of IG-BR with MCS techniques provided simulated expected train delay and predicted delays which turned the solution approach into stochastic, called S-IG-BR. We proposed using S-IG-BR because (i) IG-BR had success in solving the deterministic RSP as discussed in Chapter 3 and (ii) MCS was more effective for simulating expected values.

The stochastic optimisation model was based on the formulation from Chapter 3 with the main objective of minimising expected total train delay and then adding a random value into the uncertainty variable. For S-BR-IG procedures, MCS was added into RSP by simulating an expected value for late departure time, running time and dwell time delay. After generating the expected delays, the IG-BR provided a reschedule with the aim to minimum expected total train delays. The MCS used the mean and variance values of test instances to generate the expected solutions.

The data for experimentation were collected from Southeastern operating train company, UK (Khosravi, 2013) and State Railway of Thailand. For computational experimentation, two types of delays were used: (i) minor and general disruption and (ii) major disruption considered for the UK case study. Historical data were used as the test instances for the Thailand case study. Moreover, C# was used for experimentation with this problem. The average result of BDS-Stoch which transferred the best deterministic solution to the stochastic from Chapter 5 using the MCS process was used for analysis and comparison.

The S-IG-BR algorithm provided a good solution with 400 seconds of computation time. By comparing with BDS-Stoch, the expected total train delays of S-IG-BR were lower and more reliable than BDS-Stoch. The result showed that it was beneficial to generate estimate expected train delays which lead to reduction in total train delays. In the next chapter, we develop an S-BR-IG-LS algorithm to make a fair comparison between IG-BR and BR-IG-LS algorithms.

Chapter 7: **Sim-Biased Randomised Iterated Greedy with Local Search (S-BR-IG-LS) for Stochastic RSP**

7.1. Introduction

In the previous chapter, results of the S-IG-BR algorithm were provided and compared with the deterministic RSP solution, showing only a small performance gap. In this chapter, we selected the most popular local search heuristics which can help to increase the solutions search space. We proposed hybridisation of MCS with the BR-IG-LS algorithm to solve the RSP that deals with a stochastic delay. The BR-IG-LS was selected because it successfully solved the deterministic RSP as shown in Chapter 4.

This chapter describes how to add MCS into the BR-IG-LS for solving a stochastic RSP using proposed solution methods namely, Sim-Biased Randomised IG with Local Search (S-BR-IG-LS). Then, we compared the experimental results of S-BR-IG-LS with results from Chapter 6 to measure the efficiency of the solution methods. The experiment was executed based on two case studies from the UK and Thailand rail networks.

Chapter 7 is structured as follows: section 7.2 presents the contributions and then the proposed S-BR-IG-LS algorithm is used to solve RSP with stochastic disruption delays as described in section 7.3. All the experimental results are shown in section 7.4, separated into two sets of case studies (UK & Thailand). Section 7.5 is the conclusion of this chapter.

7.2. Contribution

The main contribution of this chapter is the development and implementation the BR-IG-LS and MCS technique, called S-BR-IG-LS, for solving the stochastic RSP. The BR-IG-LS was used to find the solution and minimise the total delay of all trains, while MCS generated random delays to simulate a real environment of the case study. MSC also provided a fair comparison between different algorithms.

To best of our knowledge, S-BR-IG-LS has not been used to solve the stochastic RSP before, therefore we are the first to propose the S-BR-IG-LS for solving stochastic RSP.

7. 3. Proposed Sim-Biased Randomised IG with Local Search (S-BR-IG-LS) for solving RSP

Results of BR-IG-LS on the deterministic situation in Chapter 4 improved the solution of IG-BR for the deterministic RSP. Therefore, we proposed the S-BR-IG-LS to solve the stochastic RSP. This chapter added MCS to the BR-IG-LS algorithm in order to improve the solution of the stochastic RSP. The main procedure was as follows: (a) Initial solution based on given timetables in our case studies which comprises of real-world data from the UK and Thailand Rail network, (b) MCS process which randomly generates expected delays, (c) IG is used to solve the RSP by using two main phases (destruction and construction), (d) the use of biased randomised algorithm to improve the resulting solution and (e) the use of local search methods to improve the speed of solution (Algorithm 7.1).

Algorithm 7.1 : Procedure for Stochastic RSP S-BR-IG-LS Algorithm

---

1: **Procedure for Stochastic RSP S-BR-IG-LS Algorithm** (r', r^D, r^R, d, b, β)
▻ r': Initial solution
▻ r^D: Partial sequence to reconstruct
▻ r^R: Trains to reinsert
▻ d: Random chosen number of trains
▻ b: Number of blocks
▻ β: Parameter for biased randomised
2: r': *GenerateInitialSolution*(); ▻ *Initial solution*
3: iteration = 0;
4: **while** execution_time < 400 **do** ▻ *Execution time 400 seconds*
5: **for** each train in b **do** ▻ *start Monte Carlo simulation*
6: trainDuration = trainDuration + generateRandomDelay ▻ *randomly generate delay*
7: **end for**
8: **for** i **=** 1 to b **do** ▻ *run for all block*
9: set r^R = empty
10: **for** $\boldsymbol{i}$ **<-** 1 to d **do** ▻ *Destruction step*
11: r^R <- remove one node at randomly from r' and insert it in r^R;
12: r^D = r' <- the remaining set of trains
13: **end for**
14: re-ordering r^R by using β to assign a probability ▻ *Biased randomised*
15: **for j <-** 1 to d **do** ▻ *Construction step*
16: r^{ans} = ApplyLocalSearch(r^R) ▻ *Insertion Neighbourhood (LS)*
17: **if** r^{ans} met with all railway constraints **then** ▻ *check all constraints*
18: solution (r^{ans}) = calculate minimise a total delay (r^{ans})
19: **end if**
20: **end for**
21: **end for**
22: solution (r^{best}) = solution (r^{best}) + solution (r^{ans}) ▻ *Sum up all solution (r^{ans})*
23: iteration++
24: **end while**
25: averageDelay = solution (r^{best}) / iteration ▻ *Average the total delay*
26:**end procedure**

Algorithm 7.1 presents the pseudocode for stochastic RSP S-BR-IG-LS. Firstly, the process starts by creating an initial solution from the rail timetable (Pseudocode: line 2). Then, we use MCS to randomly generate the amount of the delay (Pseudocode: line 5 to 7).

After that, start to implement BR-IG-LS by adding the biased randomised algorithm at the end of the destruction phase of IG algorithm to reorder the construction set before continuing to the construction phase (Pseudocode: line 10 to line 14).

In the construction phase, we reinsert the construction set back into the remain set using the local search concept, which only reinserts in random position (Pseudocode: line 15 to line 20). Moreover, we need to check that all railway constraints are satisfied (Pseudocode: line 17 to line 19).

Following the BR-IG-LS process, the MCS provides an expected total delay (Pseudocode: line 22 to line 25).

7.4. Computational experiments

In this chapter, we proposed to evaluate the performance of the S-BR-IG-LS by comparing it with the BDS-Stoch and S-IG-BR from the previous chapter. We executed all experiment on the same personal computer with Intel core i7-4500U CPU, 1.80GHz-2.4GHz, and 8GB RAM. Moreover, we used the same test instances as described in section 3.5.1.1. (UK) and 3.5.2.1. (Thailand). All parameters and processes followed the outlines detailed in section 6.5.

7.4.1. Southeastern train company, United Kingdom

The computational experiments were compared between BDS-Stoch, S-IG-BR and S-BR-IG-LS to evaluate the performance of S-BR-IG-LS. All experiments were executed for 400 seconds which served as the real-time service update procedure that provided by SSDP.

Tables 7.1-7.4 show the solution of using BDS-Stoch, S-IG-BR and S-BR-IG-LS to solve the UK case study in the stochastic situation. In each table, all columns before the column BDS-Stoch show the detail of each instance and all delays are shown in **minutes**. Then, BDS-Stoch and S-IG-BR columns were used to present the results from the previous chapter. Moreover, S-BR-IG-LS column was used to present total train delays, reliability and the performance of S-BR-IG-LS when compared with BDS-Stoch and S-IG-BR solutions from the previous chapter. Finally, at the end of each table, we calculated an average of the total delay, reliability and RD. A positive number of RD (%) means that S-BR-IG-LS gave better performance than other methods from the previous chapter.

Block Delay	# of Block(s)	# of Train(s)	Instance	BDS-Stoch		S-IG-BR			S-BR-IG-LS			
				Delay	Reliability	Delay	Reliability	RD (%) compared with BDS-Stoch	Delay	Reliability	RD (%) compared with S-IG-BR	RD (%) compared with BDS-Stoch
5	1	3	P01	124.66	0.56	119.10	0.96	4.46	114.38	0.99	3.96	8.25
	2	5	P02	159.88	0.72	155.35	0.98	2.84	154.05	0.98	0.83	3.65
10	1	3	P03	176.79	0.78	173.78	0.97	1.70	172.71	0.97	0.61	2.31
	2	5	P04	285.86	0.74	284.46	0.97	0.49	283.51	0.98	0.33	0.82
15	1	3	P05	266.81	0.57	266.48	0.96	0.12	265.16	0.98	0.50	0.62
	2	5	P06	316.62	0.84	316.45	1.00	0.05	316.44	0.95	0.00	0.06
20	1	9	P07	1133.35	0.78	1129.02	0.99	0.38	1088.16	0.96	3.62	3.99
	2	12	P08	771.00	0.59	733.03	0.87	4.92	719.13	0.88	1.90	6.73
25	1	9	P09	1409.90	0.78	1408.45	0.92	0.10	1383.40	0.88	1.78	1.88
	2	12	P10	1871.83	0.59	1871.06	0.88	0.04	1866.60	0.87	0.24	0.28
30	1	9	P11	1108.30	0.77	1092.59	0.87	1.42	1079.62	0.96	1.19	2.59
	2	12	P12	1950.01	0.73	1903.46	0.87	2.39	1888.02	0.87	0.81	3.18
Avg Delay				797.92	0.70	787.77	0.94	1.27	**777.60**	**0.94**	**1.29**	**2.55**

Table 7.1 : Stochastic Longer running/dwell times - minor and general disruption: S-BR-IG-LS Results

Table 7.1 shows the S-BR-IG-LS results of stochastic RSP considered for minor and general delays for longer running time and dwell time. The best solution for each instance is indicated in **bold**. This table shows the comparison between BDS-Stoch, S-IG-BR and S-BR-IG-LS. Results showed that S-BR-IG-LS reduced the expected total train delay at 1.29% lower than S-IG-BR and 2.55% lower than BDS-Stoch. In terms of the solution reliability, S-IG-BR and S-BR-IG-LS gave the same level at 94%, however this was 24% higher compared to the BDS-Stoch.

Late Departure	# of Train(s)	Instance	BDS-Stoch		S-IG-BR			S-BR-IG-LS			
			Delay	Reliability	Delay	Reliability	RD (%) compared with BDS-Stoch	Delay	Reliability	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR
5	1	R01	67.41	0.52	67.14	0.99	0.40	66.15	0.99	1.87	1.47
	3	R02	67.89	0.77	66.37	0.99	2.23	63.44	0.98	6.55	4.42
	5	R03	82.51	0.81	76.83	0.98	6.88	77.20	0.98	6.43	-0.48
10	1	R04	78.32	0.64	75.07	0.99	4.15	76.99	1.00	1.70	-2.55
	3	R05	94.25	0.59	91.54	0.99	2.89	89.30	0.99	5.26	2.45
	5	R06	157.85	0.73	148.17	0.98	6.14	148.08	0.98	6.19	0.06
15	1	R07	76.75	0.83	70.46	0.99	8.20	68.57	0.98	10.67	2.69
	3	R08	132.96	0.88	123.80	0.98	6.89	123.40	0.99	7.19	0.32
	5	R09	190.61	0.79	184.97	0.98	2.96	183.73	0.98	3.61	0.67
20	6	R10	208.98	0.64	190.39	0.98	8.90	190.71	0.99	8.74	-0.17
	9	R11	272.86	0.78	256.13	0.97	6.13	252.71	0.96	7.38	1.33
	12	R12	383.79	0.62	354.18	0.95	7.71	354.69	0.96	7.58	-0.15
25	6	R13	261.57	0.66	260.62	0.98	0.36	254.33	0.99	2.77	2.41
	9	R14	315.59	0.60	305.60	0.96	3.17	304.58	0.95	3.49	0.33
	12	R15	431.69	0.74	415.41	0.93	3.77	413.00	0.93	4.33	0.58
Avg Delay			188.20	0.71	179.11	0.98	4.83	**177.79**	**0.98**	**5.53**	**0.74**

Table 7.2 : Stochastic Late departures - minor and general disruption: S-BR-IG-LS Results

Table 7.2 shows S-BR-IG-LS results of stochastic RSP considered for minor and general delays for late departure times. The best solution in each instance is indicated in **bold**. The experimental results showed that S-BR-IG-LS decreased the expected total train delays when compared to the other solution methods by 0.74% better than S-IG-BR and 5.53% better than BDS-Stoch. However, if we only focus on the reliability of the solution, S-BR-IG-LS and S-IG-BR have equal level of reliability at 98%, which is 27% more reliable than BDS-Stoch.

Block Delay	# of Block(s)	Instance	BDS-Stoch		S-IG-BR			S-BR-IG-LS			
			Delay	Reliability	Delay	Reliability	RD (%) compared with BDS-Stoch	Delay	Reliability	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR
40	4	HP01	8112.99	0.33	7949.56	0.62	2.01	7864.38	0.46	3.06	1.07
	5	HP02	8108.44	0.23	7837.70	0.98	3.34	7663.19	0.64	5.49	2.23
	6	HP03	7374.15	0.57	7352.63	0.48	0.29	7336.43	0.97	0.51	0.22
50	4	HP04	9011.98	0.45	8760.50	0.54	2.79	8586.35	0.48	4.72	1.99
	5	HP05	9539.04	0.42	9239.39	0.71	3.14	9192.46	0.41	3.63	0.51
	6	HP06	7436.10	0.31	7424.27	0.96	0.16	7411.61	0.52	0.33	0.17
60	4	HP07	10356.47	0.49	10280.29	0.48	0.74	10215.74	0.56	1.36	0.63
	5	HP08	9120.18	0.20	8905.48	0.57	2.35	8856.10	0.97	2.90	0.55
	6	HP09	9551.90	0.52	9306.72	0.75	2.57	9176.88	0.88	3.93	1.40
70	4	HP10	14344.90	0.35	13908.34	0.80	3.04	13660.62	0.90	4.77	1.78
	5	HP11	11010.32	0.39	10795.17	0.74	1.95	10703.24	0.67	2.79	0.85
	6	HP12	13558.67	0.22	13384.99	0.72	1.28	13321.67	0.83	1.75	0.47
80	4	HP13	15505.31	0.60	15421.20	0.89	0.54	15301.73	0.83	1.31	0.77
	5	HP14	14868.93	0.35	14690.12	0.75	1.20	14612.04	0.96	1.73	0.53
	6	HP15	16221.56	0.50	16110.78	0.69	0.68	16102.28	0.66	0.74	0.05
Avg Delay			10941.40	0.40	10757.81	0.71	1.68	**10666.98**	**0.72**	**2.51**	**0.84**

Table 7.3 : Stochastic Longer running/dwell times - major disruption: S-BR-IG-LS

Table 7.3 presents the S-BR-IG-LS results of stochastic RSP considered for major delays for longer running time and dwell times. The best solution for each instance is indicated in **bold**. The lowest average of the total train delays was recorded by S-BR-IG-LS and results were better than S-IG-BR by 0.84% and better than BDS-Stoch by 2.51%. The reliability level of S-BR-IG-LS was also 22% higher than BDS-Stoch with only a small gap of 1% higher than S-IG-BR. The readability of each method was 72% for S-BR-IG-LS, 71% for S-IG-BR and 40% for BDS-Stoch.

Block Delay	# of Block(s)	Departure Delay	# of Train(s)	Instance	BDS-Stoch		S-IG-BR			S-BR-IG-LS			
					Delay	Reliability	Delay	Reliability	RD (%) compared with BDS-Stoch	Delay	Reliability	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR
40	4	10	3	PR01	9306.78	0.45	8236.26	0.75	11.50	8132.13	0.81	12.62	1.26
			5	PR02	8879.46	0.26	8652.64	0.52	2.55	8597.31	0.44	3.18	0.64
		20	9	PR03	7923.36	0.56	7643.25	0.60	3.54	7601.73	0.98	4.06	0.54
			12	PR04	8096.74	0.52	7905.48	0.59	2.36	7862.54	0.47	2.89	0.54
40	6	10	3	PR05	7225.57	0.41	7180.96	0.71	0.62	7137.85	0.66	1.21	0.60
			5	PR06	8720.84	0.33	8717.83	0.45	0.03	8627.82	0.55	1.07	1.03
		20	9	PR07	6667.70	0.49	6666.19	0.98	0.02	6664.65	0.44	0.05	0.02
			12	PR08	7755.11	0.50	7735.50	0.96	0.25	7542.54	0.94	2.74	2.49
50	4	10	3	PR09	11441.82	0.18	10322.71	0.42	9.78	10266.19	0.78	10.27	0.55
			5	PR10	10744.84	0.54	10406.84	0.55	3.15	10057.87	0.94	6.39	3.35
		20	9	PR11	7884.00	0.31	7814.99	0.52	0.88	7744.64	0.68	1.77	0.90
			12	PR12	10518.84	0.35	10415.75	0.60	0.98	10404.50	0.42	1.09	0.11
50	6	10	3	PR13	8605.79	0.22	8532.53	0.66	0.85	8522.55	0.79	0.97	0.12
			5	PR14	9827.53	0.56	9819.17	0.80	0.09	9816.59	0.51	0.11	0.03
		20	9	PR15	8829.11	0.21	8701.99	0.57	1.44	8605.96	0.62	2.53	1.10
			12	PR16	9693.25	0.21	9141.21	0.67	5.70	9040.39	0.56	6.74	1.10
Avg Delay					8882.55	0.38	8618.33	0.65	2.97	**8539.08**	**0.66**	**3.87**	**0.92**

Table 7.4 : Stochastic Longer running/dwell times and Late departures - major disruption: S-BR-IG-LS Results

Table 7.4 indicates the S-BR-IG-LS results of stochastic RSP considered on major disruption on late departure times. The best solution for each instance is indicated in **bold**. A comparison between S-BR-IG-LS and the other algorithms showed that the average total delay of S-BR-IG-LS was 0.92% better than S-IG-BR and 3.87% better than BDS-Stoch. The reliability coefficient showed that S-BR-IG-LS had highest reliability at 66% which was 28% higher than BDS-Stoch and 1% higher than S-IG-BR.

The SSDP separated disruptions into 3 types depending on the size as minor disruptions, general disruptions and major disruptions. Thus, we discussed our results based on the disruption types of each instance.

Disruption Type		BDS-Stoch		S-IG-BR			S-BR-IG-LS			
		Delay	Reliability	Delay	Reliability	RD (%) compared with BDS-Stoch	Delay	Reliability	RD (%) compared with S-IG-BR	RD (%) compared with BDS-Stoch
Minor and General	Longer Running/dwell times	797.92	0.70	787.77	0.94	1.27	777.60	0.94	1.29	2.55
	Late Departures	188.20	0.71	179.11	0.98	4.83	177.79	0.98	0.74	5.53
Average Delay		493.06	0.71	483.44	0.96	1.95	**477.70**	**0.96**	**1.19**	**3.12**

Table 7.5 : Summary of Experimental result of S-BR-IG-LS - Stochastic (UK) - Small

Disruption Type		BDS-Stoch		S-IG-BR			S-BR-IG-LS			
		Delay	Reliability	Delay	Reliability	RD (%) compared with BDS-Stoch	Delay	Reliability	RD (%) compared with S-IG-BR	RD (%) compared with BDS-Stoch
Major	Longer Running/dwell times	10941.40	0.40	10757.81	0.71	1.68	10666.98	0.72	0.84	2.51
	Late Departures & Longer Running/dwell times	8882.55	0.38	8618.33	0.65	2.97	8539.08	0.66	0.92	3.87
Average Delay		9911.97	0.39	9688.07	0.68	2.26	**9603.03**	**0.69**	**0.88**	**3.12**

Table 7.6 : Summary of Experimental result of S-BR-IG-LS - Stochastic (UK) - Large

Tables 7.5 and 7.6 show the summary results of small and large disruption instances. For small disruptions, S-BR-IG-LS reduced the expected total train delay by 1.19% better than S-IG-BR and 3.12% better than DS-Stoch. Moreover, for large disruptions, S-BR-IG-LS outperformed S-IG-BR by 0.88% and BDS-Stoch by 3.12%.

A comparison between small and large disruptions on the reliability showed that the size of disruptions affected the reliability level. Decreasing reliability level between small and large cases were 71% to 39% in BDS-Stoch, 96% to 68% in S-IG-BR and 96% to 69% in S-BR-IG-LS. This means that high levels of disruptions resulted in decreasing reliability.

Disruption Type		BDS-Stoch		S-IG-BR			S-BR-IG-LS			
		Delay	Reliability	Delay	Reliability	RD (%) compared with BDS-Stoch	Delay	Reliability	RD (%) compared with S-IG-BR	RD (%) compared with BDS-Stoch
Minor and General	Longer Running/dwell times	797.92	0.70	787.77	0.94	1.27	777.60	0.94	1.29	2.55
	Late Departures	188.20	0.71	179.11	0.98	4.83	177.79	0.98	0.74	5.53
Major	Longer Running/dwell times	10941.40	0.40	10757.81	0.71	1.68	10666.98	0.72	0.84	2.51
	Late Departures & Longer Running/dwell times	8882.55	0.38	8618.33	0.65	2.97	8539.08	0.66	0.92	3.87
Average Delay		5202.52	0.55	5085.76	0.82	2.24	**5040.36**	**0.82**	**0.89**	**3.12**

Table 7.7 : Summary of Experimental result of S-BR-IG-LS - Stochastic (UK) - All

Table 7.7 presents a summary of all results in section 7.4.1 and shows that CPLEX was effective for solving minor and general disruption RSPs. The S-BR-IG-LS was the best choice for major disruption RSPs and reduced the average total delay solution by 0.89% better than S-IG-BR and 3.12% better than BDS-Stoch. Moreover, S-BR-IG-LS also had the same level reliability as S-IG-BR at 82%, which was 27% higher reliability than BDS-Stoch. The results also showed that S-BR-IG-LS was more efficient than S-IG-BR with equal reliability level. This means that biased randomisation improved efficiency of the IG-BR algorithm.

7.4.2. State Railway of Thailand

Table 7.8 shows the experimental results of BDS-Stoch, S-IG-BR and S-BR-IG-LS. The first two columns show the detail of each instance. After that, BDS-Stoch, S-IG-BR and S-BR-IG-LS columns provided the results of total delay and reliability. Moreover, the last two columns provided a comparison between S-BR-IG-LS with BDS-Stoch and S-IG-BR as RD (%). Finally, at the bottom of the table, we calculated an average of the total delay and RD to provide a clearer understanding. All delays were presented in **minutes** and a positive number of RD showed that the proposed method provided an improvement, otherwise there was no improvement.

Date	Instance	BDS-Stoch		S-IG-BR		S-BR-IG-LS			
		Delay	Reliability	Delay	Reliability	Delay	Reliability	RD (%) compared with S-IG-BR	RD (%) compared with BDS-Stoch
1/2/2019	THAF101	2140.26	0.46	2132.87	0.63	1922.13	0.63	9.88	10.19
2/2/2019	THAF102	2222.09	0.43	2141.47	0.77	2187.60	0.83	-2.15	1.55
3/2/2019	THAF103	2069.37	0.33	2046.06	0.91	1801.55	0.63	11.95	12.94
4/2/2019	THAF104	1547.01	0.26	1368.37	0.81	1249.54	0.83	8.68	19.23
5/2/2019	THAF105	2289.36	0.24	2284.32	0.72	2262.59	0.70	0.95	1.17
6/2/2019	THAF106	1665.72	0.52	1618.95	0.69	1522.68	0.83	5.95	8.59
7/2/2019	THAF107	2637.56	0.20	2500.98	0.88	2466.71	0.60	1.37	6.48
8/2/2019	THAF108	2768.12	0.68	2692.45	0.71	2561.55	0.69	4.86	7.46
9/2/2019	THAF109	2571.86	0.67	1984.98	0.61	1845.15	0.66	7.04	28.26
10/2/2019	THAF110	1643.44	0.21	1520.22	0.74	1517.03	0.74	0.21	7.69
11/2/2019	THAF111	1855.58	0.70	1826.68	0.67	1774.14	0.75	2.88	4.39
12/2/2019	THAF112	1981.48	0.53	1951.20	0.61	1884.80	0.81	3.40	4.88
13/2/2019	THAF113	2101.69	0.32	1807.63	0.66	1774.24	0.68	1.85	15.58
14/2/2019	THAF114	2117.11	0.33	1944.44	0.73	1913.95	0.67	1.57	9.60
15/2/2019	THAF115	2866.43	0.53	2645.20	0.68	2615.02	0.81	1.14	8.77
16/2/2019	THAF116	3026.04	0.55	2797.11	0.81	2763.69	0.64	1.19	8.67
17/2/2019	THAF117	2285.48	0.65	2273.10	0.93	2275.90	0.90	-0.12	0.42
18/2/2019	THAF118	2583.71	0.26	2570.77	0.62	2472.62	0.64	3.82	4.30
19/2/2019	THAF119	1689.85	0.65	1537.25	0.75	1472.98	0.92	4.18	12.83
20/2/2019	THAF120	1854.07	0.60	1829.25	0.79	1769.53	0.81	3.26	4.56
21/2/2019	THAF121	3916.94	0.44	3212.71	0.68	3200.53	0.60	0.38	18.29
22/2/2019	THAF122	3993.31	0.51	3794.86	0.77	3770.80	0.84	0.63	5.57
23/2/2019	THAF123	3767.96	0.27	3460.03	0.65	3393.42	0.78	1.93	9.94
24/2/2019	THAF124	3055.96	0.48	3051.14	0.87	3024.50	0.82	0.87	1.03
25/2/2019	THAF125	3561.76	0.22	3400.97	0.78	3392.38	0.72	0.25	4.76
26/2/2019	THAF126	5145.14	0.52	5091.63	0.71	5090.09	0.83	0.03	1.07
27/2/2019	THAF127	3595.63	0.36	3283.33	0.90	3224.73	0.98	1.79	10.32
28/2/2019	THAF128	3207.96	0.53	2969.39	0.89	2976.17	0.73	-0.23	7.23
Average Delay		2648.60	0.44	2490.62	0.75	**2433.07**	**0.75**	**2.31**	**8.14**

Table 7.8 : Stochastic Thai: S-BR-IG-LS

Table 7.8 shows the results of stochastic RSP on the Thailand case study. The best solution for each instance is indicated in **bold**. Results showed that S-BR-IG-LS reduced the expected total delays at 2.31% lower than S-IG-BR and 8.14% lower than BDS-Stoch. For the reliability, the results showed that S-IG-BR had higher reliability than the BDS-Stoch by 31%. This high reliability suggested that the solution methods were more flexible and suitable for solving the real-world case study. Moreover, S-IG-BR and S-BR-IG-LS gave equal reliability at 75%.

7.5. Chapter Conclusion

This chapter proposed BR-IG-LS with MCS techniques to provide simulated expected train delay and predicted delays. This turned the solution approach into stochastic, called S-BR-IG-LS. We proposed using S-BR-IG-LS because (i) LS had previous success in improving the solution quality of IG-BR for the deterministic RSP as discussed in Chapter 4 and (ii) MCS is more effective for simulating expected values.

For S-BR-IG-LS procedures, MCS was added into RSP by simulating an expected value for late departure time, running time and dwell delay into the initial solution. After that, the BR-IG-LS was shown to improve the rescheduling process by using the stochastic optimisation model based on the formulation from the previous chapter, with the aim to minimum expected total train delays. The MCS used the mean and variance values of test instances to generate the expected solutions.

Data for experimentation were retrieved from the Southeastern operating train company, UK (Khosravi, 2013) and State Railway of Thailand. For computational experimentation, two types of delay were used: (i) minor and general disruption and (ii) major disruption, considered for the UK case study, while historical data were used as test instances for the Thailand case study. Moreover, C# was used for experimentation with this problem. The average result from Chapter 6 was used for analysis and comparison.

The S-BR-IG-LS algorithm provided a better performance than S-IG-BR in 400 seconds of computation time. Moreover, the expected total train delays of S-BR-IG-LS were lower than BDS-Stoch. Results showed that it was beneficial to generate an estimated expected total train delay which then led to reduction in total train delays and increased the reliability. In the next chapter, we develop a S-BR-VNS to make a fair comparison between all solution methods proposed in this thesis.

Chapter 8: Sim-Biased Randomised Variables Neighbourhood Search (S-BR-VNS) for Stochastic RSP

8.1. Introduction

In the previous chapter, results of S-BR-IG-LS algorithm showed good performance on the stochastic RSP but provided only a small gap better than S-IG-BR algorithms. Therefore, we chose the metaheuristic method which discussed in Chapter 5, which outperform in the deterministic RSP, to solve the stochastic RSP and further improve the results.

This chapter described how to add MCS into the VNS or BR-VNS for solving a stochastic RSP using the proposed solution methods namely, Sim-VNS (S-VNS) or Sim-Biased Randomised VNS (S-BR-VNS). Then, we compare the experimental results of the proposed solution methods with results from the previous chapter to measure their efficiency. The experiment was executed based on two case studies from the UK and Thailand rail networks.

The chapter is structured as follows: section 8.2 presents the contributions and then the proposed S-BR-VNS algorithm, used to solve RSP with stochastic disruption delays is described in section 8.3. All the experiment results are shown in section 8.4, separated into two sets of case studies (UK & Thailand). Section 8.5 is the conclusion.

8.2. Contribution

The main contribution of this chapter is the development and implementation the VNS or BR-VNS and MCS technique, called S-VNS or S-BR-VNS, for solving the stochastic RSP. The VNS and BR-IG-VNS were used to find the solution and minimise the total delay of all trains, while MCS generated random delays to simulate a real environment of the case study. Moreover, the MSC also provide a fair comparison between different algorithms.

According to the literature review, no one has used MCS and VNS or BR-VNS algorithm as sim-optimisation to solve the stochastic RSP before; therefore, this is the first study proposing S-VNS or S-BR-VNS to solve the RSP.

8.3. Proposed Sim-Biased Randomised VNS (S-BR-VNS) for solving RSP

Results of BR-VNS in Chapter 5 outperformed other methods for the deterministic RSP; therefore, we proposed the S-BR-VNS to solve the stochastic RSP. In this chapter, we proposed the S-VNS and S-BR-VNS algorithms to solve the stochastic RSP by combined VNS and BR-VNS with the MCS.

The different between S-VNS and S-BR-VSN is the initial solution. For S-VNS, we used the original timetable as the initial solution and then used the VNS to improve the solution. For the S-BR-VNS, we started by using the biased randomised algorithm to improve the initial solution and then solved it by using the VNS algorithm.

The process of implementation of the S-BR-VNS is described as follows: (a) select the set of neighbourhood structures which consider the rail network characteristics, (b) select the initial solution of timetable which is generated by using a biased randomised algorithm to improve the initial solution of the case studies, (c) use the MCS technique to randomly generate some delays into the system and (d) use VNS to improve the solution by using three main phases (shaking, local search and move or not) (Algorithm 8.1).

Algorithm 8.1 : Procedure for Stochastic RSP S-BR-VNS Algorithm

---

```
1: Procedure Stochastic RSP S-BR-VNS Algorithm (x, b, β)
                                                              ▻ x: Initial solution
                                                              ▻ b: Number of blocks
                                                              ▻ β: Parameter: biased randomised
2: set of neighbourhood structures N^k; k = 1, ..., k^max     ▻ set neighbourhood structures
3:  for i = 1 to b do                                         ▻ run for all block
4:     using β probability to assign order of train in each block  ▻ Biased randomised
5:  end for
6:  x: GenerateBiasedRandomisedInitialSolution();             ▻ Initial solution
7:  k = 1                                                     ▻ start from first neighbourhood structure
8:  iteration = 0;
9:  while execution_time < 400 do                             ▻ Execution time 400 seconds
10:     for each train in b do                                ▻ start Monte Carlo simulation
11:        trainDuration = trainDuration + generateRandomDelay ▻ randomly generate delay
12:     end for
13:     for k = 1 to k^max do                                 ▻ k^max=4; four neighbourhood structures
14:         select a random x' from N_k(x)                    ▻ shaking
15:        if x' met with all railway constraints then        ▻ check all constraints
16:            solution (x') = calculate minimise a total delay (x')
17:        end if
18:        x'' = neighbourhood search of x'                   ▻ local search
19:        if x'' met with all railway constraints then       ▻ check all constraints
20:            solution (x'') = calculate minimise a total delay (x'')
21:        end if
22:        if solution (x'') < solution (x) then              ▻ move or not
23:          x = x''
24:          k = 1                                            ▻ return to first neighbourhood structure
25:        else if
26:          k = k + 1                                        ▻ move to next neighbourhood structure
27:        end if
28:     end for
29:     solution (x^best) = solution (x^best) + solution (x)  ▻ Sum up all solution (r^ans)
30:     iteration++                                           ▻ Count number of iterations
31:  end while
32:  averageDelay = solution (x^best) / iteration             ▻ Average the total delay
33: end procedure
```

Algorithm 8.1 presents the pseudocode for stochastic RSP S-BR-VNS Algorithm. Firstly, the process starts by selecting the set of neighbourhood structure (Pseudocode: line 2). Secondly, we use the biased randomised algorithm to assign the probability to reorder the train sequence and create an initial solution (Pseudocode: line 3 to line 6). Then, we set to repeat the whole process until a stopping condition is met (Pseudocode: line 9). We consider the stochastic disruption situation, so the expected delay is generated by using the MCS (Pseudocode: line 10 to 12).

Then, we start to implement the VNS from the first neighbourhood structure until the last neighbourhood structure (Pseudocode: line 13). In each neighbourhood, we perform the shaking phase, local search phase and move or not phase. For the shaking phase of VNS, we randomly select an *x'* solution from the initial solution *x* in order to avoid cycling (Pseudocode: line 14 to line 17). Next, in the local search phase, we use the k^{th} neighbourhood rule to find a new solution *x"* (Pseudocode: line 18 to line 21). After that, the move or not phase is applied to check if the new solution *x"* is better than the best solution *x*, set best solution *x* = new solution *x"* and continue the search on the first neighbourhood structure N_k, (k=1); otherwise move to use next neighbourhood structure N_k, (k=k+1) (Pseudocode: line 22 to line 27). Finally, the MCS process is used to find an expected average of the total delay (Pseudocode: line 29 to line 32). In addition, at the shaking and local search phases, we also required the process to confirm that all railway operational constraints were still valid (Pseudocode: line 15 to line 17 and line 19 to line 21).

8.4. Computational experiments

In this chapter, we proposed to evaluate the performance of the S-BR-VNS by comparing it to the BDS-Stoch, S-IG-BR and S-BR-IG-LS from Chapter 7. We executed all experiments on the same personal computer with Intel core i7-4500U CPU, 1.80GHz-2.4GHz, and 8GB RAM. Moreover, we used the same test instances as described in section 3.5.1.1. (UK) and 3.5.2.1. (Thailand). All parameters and processes follow the outlines detailed in section 6.5.

8.4.1. Southeastern train company, United Kingdom

The S-BR-VNS was compared to the result from the previous chapter and the basic S-VNS to evaluate the performance and reliability of the BR-VNS.

Tables 8.1, 8.3, 8.5, 8.7, 8.9, 8.11 and 8.13 show the experimental results using BDS-Stoch, S-IG-BR, S-BR-IG-LS, S-VNS and S-BR-VNS. In each table, all columns before the column BDS-Stoch show details of each instance. Moreover, in each solution method column shows the expected total train delay and reliability. Finally, at the bottom row of the table, we calculated an average of total delay, and the reliability level.

Tables 8.2, 8.4, 8.6, 8.8, 8.10, 8.12 and 8.14 compares the average delay between each solution method from Tables 8.1, 8.3, 8.5, 8.7, 8.9, 8.11 and 8.13 for comparison. Each sub column under the name of each solution shows Relative Deviation (RD) with compared to all other solution methods proposed in this thesis as CPLEX (RD (%) compared with CPLEX), IG-BR (RD (%) compared with IG-BR), BR-IG-LS (RD (%) compared with BR-IG-LS) and VNS (RD (%) compared with VNS).

In addition, all delays time are presented in **minutes** and a positive number of RD (%) means that the method had a better performance than the other methods, while a negative number shows as no improvement.

Block Delay	# of Block(s)	# of Train(s)	Instance	BDS-Stoch		S-IG-BR		S-BR-IG-LS		S-VNS		S-BR-VNS	
				Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability
5	1	3	P01	124.66	0.56	119.10	0.96	114.38	0.99	111.79	0.98	108.83	0.98
	2	5	P02	159.88	0.72	155.35	0.98	154.05	0.98	153.44	0.99	146.76	1.00
10	1	3	P03	176.79	0.78	173.78	0.97	172.71	0.97	172.68	0.96	171.01	0.96
	2	5	P04	285.86	0.74	284.46	0.97	283.51	0.98	283.68	0.99	282.11	0.98
15	1	3	P05	266.81	0.57	266.48	0.96	265.16	0.98	264.67	0.98	262.38	0.99
	2	5	P06	316.62	0.84	316.45	1.00	316.44	0.95	315.56	0.95	315.37	0.99
20	1	9	P07	1133.35	0.78	1129.02	0.99	1088.16	0.96	1071.66	0.99	1032.37	0.98
	2	12	P08	771.00	0.59	733.03	0.87	719.13	0.88	713.44	0.85	669.69	0.92
25	1	9	P09	1409.90	0.78	1408.45	0.92	1383.40	0.88	1371.32	0.99	1291.66	0.95
	2	12	P10	1871.83	0.59	1871.06	0.88	1866.60	0.87	1842.78	0.92	1827.94	0.91
30	1	9	P11	1108.30	0.77	1092.59	0.87	1079.62	0.96	1076.14	0.87	1022.81	0.93
	2	12	P12	1950.01	0.73	1903.46	0.87	1888.02	0.87	1839.00	0.93	1813.82	0.91
Avg Delay				797.92	0.70	787.77	0.94	777.60	0.94	768.01	0.95	**745.40**	**0.96**

Table 8.1 : Stochastic Longer running/dwell times - minor and general disruption: S-BR-VNS Results

Block Delay	# of Block(s)	# of Train(s)	Instance	S-VNS			S-BR-VNS			
				RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with S-VNS
5	1	3	P01	10.32	6.13	2.26	12.70	8.62	4.85	2.65
	2	5	P02	4.03	1.23	0.40	8.21	5.53	4.73	4.35
10	1	3	P03	2.32	0.63	0.02	3.27	1.60	0.99	0.97
	2	5	P04	0.76	0.27	-0.06	1.31	0.83	0.49	0.55
15	1	3	P05	0.80	0.68	0.19	1.66	1.54	1.05	0.86
	2	5	P06	0.33	0.28	0.28	0.39	0.34	0.34	0.06
20	1	9	P07	5.44	5.08	1.52	8.91	8.56	5.13	3.67
	2	12	P08	7.47	2.67	0.79	13.14	8.64	6.88	6.13
25	1	9	P09	2.74	2.64	0.87	8.39	8.29	6.63	5.81
	2	12	P10	1.55	1.51	1.28	2.34	2.30	2.07	0.81
30	1	9	P11	2.90	1.51	0.32	7.71	6.39	5.26	4.96
	2	12	P12	5.69	3.39	2.60	6.98	4.71	3.93	1.37
Avg RD(%)				3.75	2.51	1.23	**6.58**	**5.38**	**4.14**	**2.94**

Table 8.2 : Stochastic Longer running/dwell times - minor and general disruption: S-BR-VNS Results (RD)

Tables 8.1 and 8.2 show S-BR-VNS results of stochastic RSP considered on minor and general delays for longer running time and dwell time. The best solution for each instance is indicated in **bold**. Table 8.1, S-BR-VNS provided highest reliability at 96% with only 1% better than S-VNS and 2% better than S-IG-BR and S-BR-IG-LS. Moreover, the BDS-Stoch had lowest reliability at 70%. In terms of expected total train delays, Table 8.2

provided a fair comparison between BDS-Stoch, S-IG-BR, S-BR-IG-LS, S-VNS and S-BR-VNS. Results showed that S-BR-VNS had lower expected total train delay than the other solution methods at 2.94% better than S-VNS, 4.14% better than S-BR-IG-LS, 5.38% better than S-IG-BR and 6.58% better than BDS-Stoch. Therefore, S-BR-VNS had the best performance in terms of average total delay and reliability. The results also showed that the biased randomised heuristic improve the performance of the basis VNS.

Late Departure	# of Train(s)	Instance	BDS-Stoch		S-IG-BR		S-BR-IG-LS		S-VNS		S-BR-VNS	
			Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability
5	1	R01	67.41	0.52	67.14	0.99	66.15	0.99	64.47	0.95	64.37	0.95
	3	R02	67.89	0.77	66.37	0.99	63.44	0.98	62.07	0.99	61.84	1.00
	5	R03	82.51	0.81	76.83	0.98	77.20	0.98	74.09	0.97	74.47	0.96
10	1	R04	78.32	0.64	75.07	0.99	76.99	1.00	71.29	0.99	70.83	1.00
	3	R05	94.25	0.59	91.54	0.99	89.30	0.99	88.33	0.99	82.46	0.99
	5	R06	157.85	0.73	148.17	0.98	148.08	0.98	146.64	0.96	142.23	0.96
15	1	R07	76.75	0.83	70.46	0.99	68.57	0.98	67.77	0.97	67.86	0.98
	3	R08	132.96	0.88	123.80	0.98	123.40	0.99	123.13	0.98	120.44	1.00
	5	R09	190.61	0.79	184.97	0.98	183.73	0.98	181.77	1.00	177.34	1.00
20	6	R10	208.98	0.64	190.39	0.98	190.71	0.99	189.05	0.99	186.88	0.97
	9	R11	272.86	0.78	256.13	0.97	252.71	0.96	246.78	0.97	241.62	0.95
	12	R12	383.79	0.62	354.18	0.95	354.69	0.96	350.21	0.95	342.26	1.00
25	6	R13	261.57	0.66	260.62	0.98	254.33	0.99	250.32	1.00	239.04	0.98
	9	R14	315.59	0.60	305.60	0.96	304.58	0.95	303.56	0.99	285.01	0.97
	12	R15	431.69	0.74	415.41	0.93	413.00	0.93	390.48	0.99	381.99	0.96
Avg Delay			188.20	0.71	179.11	0.98	177.79	0.98	174.00	0.98	**169.24**	**0.98**

Table 8.3 : Stochastic Late departures - minor and general disruption: S-BR-VNS Results

Late Departure	# of Train(s)	Instance	S-VNS			S-BR-VNS			
			RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with S-VNS
5	1	R01	4.37	3.98	2.55	4.51	4.13	2.69	0.15
	3	R02	8.57	6.48	2.16	8.91	6.83	2.52	0.37
	5	R03	10.20	3.57	4.03	9.74	3.07	3.54	-0.51
10	1	R04	8.98	5.04	7.40	9.56	5.65	8.00	0.64
	3	R05	6.29	3.50	1.08	12.52	9.92	7.66	6.65
	5	R06	7.11	1.03	0.98	9.90	4.00	3.95	3.00
15	1	R07	11.71	3.82	1.17	11.59	3.70	1.04	-0.13
	3	R08	7.39	0.54	0.22	9.42	2.71	2.40	2.18
	5	R09	4.64	1.73	1.07	6.96	4.12	3.48	2.43
20	6	R10	9.54	0.70	0.87	10.58	1.84	2.01	1.15
	9	R11	9.56	3.65	2.35	11.45	5.67	4.39	2.09
	12	R12	8.75	1.12	1.26	10.82	3.37	3.51	2.27
25	6	R13	4.30	3.95	1.58	8.61	8.28	6.01	4.51
	9	R14	3.81	0.67	0.33	9.69	6.74	6.42	6.11
	12	R15	9.55	6.00	5.45	11.51	8.04	7.51	2.17
Avg RD(%)			7.55	2.86	2.13	**10.07**	**5.51**	**4.81**	**2.73**

Table 8.4 : Stochastic Late departures - minor and general disruption: S-BR-VNS Results (RD)

Tables 8.3 and 8.4 show S-BR-VNS results of stochastic RSP considered on minor and general delays for late departures time. The best solution for each instance is indicated in **bold**. In Table 8.3, S-BR-VNS, S-VNS, S-BR-IG-LS and S-IG-BR had equal reliability at 98% with 27% higher than BDS-Stoch. Moreover, Table 8.4 provided a fair comparison between BDS-Stoch, S-IG-BR, S-BR-IG-LS, S-VNS and S-BR-VNS in terms of the expected total train delays. Results showed that S-BR-VNS had the best performance than the other solution methods at 2.73% better than S-VNS, 4.81% better than S-BR-IG-LS, 5.51% better than S-IG-BR and 10.07% better than BDS-Stoch. Therefore, S-BR-VNS provided the best solution in terms of average total delay. The comparison between S-BR-VNS and S-VNS was used to evaluate the efficiency of the biased randomised heuristic.

Block Delay	# of Block(s)	Instance	BDS-Stoc		S-IG-BR		S-BR-IG-LS		S-VNS		S-BR-VNS	
			Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability
40	4	HP01	8112.99	0.33	7949.56	0.62	7864.38	0.46	7789.10	0.65	7589.06	0.55
	5	HP02	8108.44	0.23	7837.70	0.98	7663.19	0.64	7636.98	0.77	7424.80	0.91
	6	HP03	7374.15	0.57	7352.63	0.48	7336.43	0.97	7338.21	0.87	7326.09	0.77
50	4	HP04	9011.98	0.45	8760.50	0.54	8586.35	0.48	8365.81	0.92	8157.44	0.61
	5	HP05	9539.04	0.42	9239.39	0.71	9192.46	0.41	9191.80	0.76	8984.89	0.82
	6	HP06	7436.10	0.31	7424.27	0.96	7411.61	0.52	7410.36	0.54	7397.69	0.97
60	4	HP07	10356.47	0.49	10280.29	0.48	10215.74	0.56	10183.91	0.77	9837.65	0.63
	5	HP08	9120.18	0.20	8905.48	0.57	8856.10	0.97	8849.81	0.87	8750.50	0.58
	6	HP09	9551.90	0.52	9306.72	0.75	9176.88	0.88	9118.39	0.55	8878.88	0.73
70	4	HP10	14344.90	0.35	13908.34	0.80	13660.62	0.90	13622.11	0.61	13405.88	0.73
	5	HP11	11010.32	0.39	10795.17	0.74	10703.24	0.67	10598.95	0.74	10385.16	0.80
	6	HP12	13558.67	0.22	13384.99	0.72	13321.67	0.83	13140.91	0.60	12575.83	0.75
80	4	HP13	15505.31	0.60	15421.20	0.89	15301.73	0.83	15132.44	0.55	14811.11	0.55
	5	HP14	14868.93	0.35	14690.12	0.75	14612.04	0.96	14592.20	0.90	14255.68	0.63
	6	HP15	16221.56	0.50	16110.78	0.69	16102.28	0.66	15709.35	0.75	15489.49	0.83
Avg Delay			10941.40	0.40	10757.81	0.71	10666.98	0.72	10578.69	0.72	**10351.34**	**0.72**

Table 8.5 : Stochastic Longer running/dwell times - major disruption: S-BR-VNS Results

Block Delay	# of Block(s)	Instance	S-VNS			S-BR-VNS			
			RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with S-VNS
40	4	HP01	3.99	2.02	0.96	6.46	4.53	3.50	2.57
	5	HP02	5.81	2.56	0.34	8.43	5.27	3.11	2.78
	6	HP03	0.49	0.20	-0.02	0.65	0.36	0.14	0.17
50	4	HP04	7.17	4.51	2.57	9.48	6.88	5.00	2.49
	5	HP05	3.64	0.52	0.01	5.81	2.75	2.26	2.25
	6	HP06	0.35	0.19	0.02	0.52	0.36	0.19	0.17
60	4	HP07	1.67	0.94	0.31	5.01	4.31	3.70	3.40
	5	HP08	2.96	0.63	0.07	4.05	1.74	1.19	1.12
	6	HP09	4.54	2.02	0.64	7.05	4.60	3.25	2.63
70	4	HP10	5.04	2.06	0.28	6.55	3.61	1.86	1.59
	5	HP11	3.74	1.82	0.97	5.68	3.80	2.97	2.02
	6	HP12	3.08	1.82	1.36	7.25	6.05	5.60	4.30
80	4	HP13	2.40	1.87	1.11	4.48	3.96	3.21	2.12
	5	HP14	1.86	0.67	0.14	4.12	2.96	2.44	2.31
	6	HP15	3.16	2.49	2.44	4.51	3.86	3.81	1.40
Avg RD(%)			3.31	1.67	0.83	**5.39**	**3.78**	**2.96**	**2.15**

Table 8.6 : Stochastic Longer running/dwell times - major disruption: S-BR-VNS Results (RD)

Tables 8.5 and 8.6 present S-BR-VNS results of stochastic RSP considered for major delays of longer running time and dwell times. The best solution for each instance is indicated in **bold**. In Table 8.5, S-BR-VNS, S-VNS and S-BR-IG-LS provided highest reliability at 72% with only 1% better than S-IG-BR. Moreover, the reliability of BDS-Stoch solution was 40% at 32% lower than highest reliability method and 31% lower than S-IG-BR. Moreover, Table 8.6 provided a fair comparison of the expected average total train delays between BDS-Stoch, S-IG-BR, S-BR-IG-LS, S-VNS and S-BR-VNS. Results showed that S-BR-VNS was more effective than the other solution methods by 2.15% better than S-VNS, 2.96% better than S-BR-IG-LS, 3.78% better than S-IG-BR and 5.39% better than BDS-Stoch.

Block Delay	# of Block(s)	Departure Delay	# of Train(s)	Instance	BDS-Stoc		S-IG-BR		S-BR-IG-LS		S-VNS		S-BR-VNS	
					Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability
40	4	10	3	PR01	9306.78	0.45	8236.26	0.75	8132.13	0.81	7922.76	0.81	7807.20	0.60
			5	PR02	8879.46	0.26	8652.64	0.52	8597.31	0.44	8023.43	0.87	7508.38	0.81
		20	9	PR03	7923.36	0.56	7643.25	0.60	7601.73	0.98	7550.27	0.77	7420.23	0.42
			12	PR04	8096.74	0.52	7905.48	0.59	7862.54	0.47	7575.08	0.68	7514.76	0.97
40	6	10	3	PR05	7225.57	0.41	7180.96	0.71	7137.85	0.66	7119.56	0.54	7115.12	0.83
			5	PR06	8720.84	0.33	8717.83	0.45	8627.82	0.55	8539.53	0.43	8534.84	0.72
		20	9	PR07	6667.70	0.49	6666.19	0.98	6664.65	0.44	6664.04	0.48	6664.04	0.45
			12	PR08	7755.11	0.50	7735.50	0.96	7542.54	0.94	7496.00	0.67	7483.10	0.44
50	4	10	3	PR09	11441.82	0.18	10322.71	0.42	10266.19	0.78	9790.37	0.85	9293.93	0.61
			5	PR10	10744.84	0.54	10406.84	0.55	10057.87	0.94	9798.85	0.41	9335.99	0.71
		20	9	PR11	7884.00	0.31	7814.99	0.52	7744.64	0.68	7725.66	0.51	7716.65	0.85
			12	PR12	10518.84	0.35	10415.75	0.60	10404.50	0.42	10400.21	0.72	10338.93	0.52
50	6	10	3	PR13	8605.79	0.22	8532.53	0.66	8522.55	0.79	8492.99	0.88	8441.79	0.45
			5	PR14	9827.53	0.56	9819.17	0.80	9816.59	0.51	9817.82	0.99	9812.04	0.57
		20	9	PR15	8829.11	0.21	8701.99	0.57	8605.96	0.62	8434.54	0.54	8308.12	0.89
			12	PR16	9693.25	0.21	9141.21	0.67	9040.39	0.56	8955.09	0.44	8866.78	0.95
Avg Delay					8882.55	0.38	8618.33	0.65	8539.08	0.66	8394.14	0.66	**8260.12**	**0.67**

Table 8.7 : Stochastic Longer running/dwell times and Late departures - major disruption: S-BR-VNS Results

Block Delay	# of Block(s)	Departure Delay	# of Train(s)	Instance	S-VNS			S-BR-VNS			
					RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with S-VNS
40	4	10	3	PR01	14.87	3.81	2.57	16.11	5.21	4.00	1.46
			5	PR02	9.64	7.27	6.68	15.44	13.22	12.67	6.42
		20	9	PR03	4.71	1.22	0.68	6.35	2.92	2.39	1.72
			12	PR04	6.44	4.18	3.66	7.19	4.94	4.42	0.80
40	6	10	3	PR05	1.47	0.85	0.26	1.53	0.92	0.32	0.06
			5	PR06	2.08	2.05	1.02	2.13	2.10	1.08	0.05
		20	9	PR07	0.05	0.03	0.01	0.05	0.03	0.01	0.00
			12	PR08	3.34	3.10	0.62	3.51	3.26	0.79	0.17
50	4	10	3	PR09	14.43	5.16	4.63	18.77	9.97	9.47	5.07
			5	PR10	8.80	5.84	2.58	13.11	10.29	7.18	4.72
		20	9	PR11	2.01	1.14	0.25	2.12	1.26	0.36	0.12
			12	PR12	1.13	0.15	0.04	1.71	0.74	0.63	0.59
50	6	10	3	PR13	1.31	0.46	0.35	1.91	1.06	0.95	0.60
			5	PR14	0.10	0.01	-0.01	0.16	0.07	0.05	0.06
		20	9	PR15	4.47	3.07	1.99	5.90	4.53	3.46	1.50
			12	PR16	7.62	2.04	0.94	8.53	3.00	1.92	0.99
Avg RD(%)					5.50	2.60	1.70	**7.01**	**4.16**	**3.27**	**1.60**

Table 8.8 : Stochastic Longer running/dwell times and Late departures - major disruption: S-BR-VNS Results (RD)

Tables 8.7 and 8.8 highlight S-BR-VNS results of stochastic RSP considered on major disruption for late departures time. In terms of expected total train delays, Table 8.7

showed that S-BR-VNS had the highest reliability at 67% with 2% better than IG-BR and only 1% better than S-VNS and S-BR-IG-LS. Moreover, the BDS-Stoch had only 38% of reliability level. Then, Table 8.8 provided a fair evaluation of the expected total train delay between BDS-Stoch, S-IG-BR, S-BR-IG-LS, S-VNS and S-BR-VNS. Results showed that S-BR-VNS provided the best solution than the other proposed methods at 1.60% better than S-VNS, 3.27% better than S-BR-IG-LS, 4.16% better than S-IG-BR and 7.01% better than BDS-Stoch.

As we mentioned before, we separated the instance sets into 2 groups depending on the size of disruption as minor and general size of disruption and major size of disruption. We summarise the results dependent on size of delay separately as follows:

Disruption Type		BDS-Stoch		S-IG-BR		S-BR-IG-LS		S-VNS		S-BR-VNS	
		Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability
Minor and General	Longer Running/dwell times	797.92	0.70	787.77	0.94	777.60	0.94	768.01	0.95	745.40	0.96
	Late Departures	188.20	0.71	179.11	0.98	177.79	0.98	174.00	0.98	169.24	0.98
Average Delay		493.06	0.71	483.44	0.96	477.70	0.96	471.00	0.97	**457.32**	**0.97**

Table 8.9 : Summary of Experimental result of S-BR-VNS - Stochastic (UK) - Small

Disruption Type		S-VNS			S-BR-VNS			
		RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with S-VNS
Minor and General	Longer Running/dwell times	3.75	2.51	1.23	6.58	5.38	4.14	2.94
	Late Departures	7.55	2.86	2.13	10.07	5.51	4.81	2.73
Average Delay		4.47	2.57	1.40	**7.25**	**5.40**	**4.27**	**2.91**

Table 8.10 : Summary of Experimental result of S-BR-VNS - Stochastic (UK) - Small

In Tables 8.9 and 8.10, the summary results of small disruption instances are shown. When we compared between BDS-Stoch, S-IG-BR, S-BR-IG-LS, S-VNS and S-BR-VNS, the best solution was provided by S-BR-VNS at 2.91% better than S-VNS, 4.27% better than S-BR-IG-LS, 5.40% better than S-IG-BR and 7.25% better than BDS-Stoch in terms of the expected total train delays. However, when we focused on the reliability, S-BR-VNS and S-VNS had highest reliability level at 97% followed by S-IG-BR and S-BR-IG-LS at 96%. Moreover, the BDS-Stoch had lowest reliability at 71%.

Disruption Type		BDS-Stoch		S-IG-BR		S-BR-IG-LS		S-VNS		S-BR-VNS	
		Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability
Major	Longer Running/dwell times	10941.40	0.40	10757.81	0.71	10666.98	0.72	10578.69	0.72	10351.34	0.72
	Late Departures & Longer Running/dwell times	8882.55	0.38	8618.33	0.65	8539.08	0.66	8394.14	0.66	8260.12	0.67
Average Delay		9911.97	0.39	9688.07	0.68	9603.03	0.69	9486.41	0.69	**9305.73**	**0.70**

Table 8.11 : Summary of Experimental result of S-BR-VNS - Stochastic (UK) - Large

Disruption Type		S-VNS			S-BR-VNS			
		RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with S-VNS
Major	Longer Running/dwell times	3.31	1.67	0.83	5.39	3.78	2.96	2.15
	Late Departures & Longer Running/dwell times	5.50	2.60	1.70	7.01	4.16	3.27	1.60
Average Delay		4.29	2.08	1.21	**6.12**	**3.95**	**3.10**	**1.90**

Table 8.12 : Summary of Experimental result of S-BR-VNS - Stochastic (UK) - Large (RD)

Tables 8.11 and 8.12 show the summary results for large disruption instances. When we compared between BDS-Stoch, S-IG-BR, S-BR-IG-LS, S-VNS and S-BR-VNS, S-BR-VNS reduced the expected total train delays 1.90% lower than S-VNS, 3.10% lower than S-BR-IG-LS, 3.95% lower than S-IG-BR and 6.12% lower than BDS-Stoch. S-BR-VNS had the highest reliability level at 70% followed by S-VNS and S-BR-IG-LS at 69%. Moreover, S-IG-BR had reliability at 68% with only 1% lower than S-VNS and S-BR-IG-LS and the BDS-Stoch had lowest reliability at 39%.

Disruption Type		BDS-Stoch		S-IG-BR		S-BR-IG-LS		S-VNS		S-BR-VNS	
		Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability
Minor and General	Longer Running/dwell times	797.92	0.70	787.77	0.94	777.60	0.94	768.01	0.95	745.40	0.96
	Late Departures	188.20	0.71	179.11	0.98	177.79	0.98	174.00	0.98	169.24	0.98
Major	Longer Running/dwell times	10941.40	0.40	10757.81	0.71	10666.98	0.72	10578.69	0.72	10351.34	0.72
	Late Departures & Longer Running/dwell times	8882.55	0.38	8618.33	0.65	8539.08	0.66	8394.14	0.66	8260.12	0.67
Average Delay		5202.52	0.55	5085.76	0.82	5040.36	0.82	4978.71	0.83	**4881.52**	**0.83**

Table 8.13 : Summary of Experimental result of S-BR-VNS - Stochastic (UK) - All

Disruption Type		S-VNS			S-BR-VNS			
		RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with S-VNS
Minor and General	Longer Running/dwell times	3.75	2.51	1.23	6.58	5.38	4.14	2.94
	Late Departures	7.55	2.86	2.13	10.07	5.51	4.81	2.73
Major	Longer Running/dwell times	3.31	1.67	0.83	5.39	3.78	2.96	2.15
	Late Departures & Longer Running/dwell times	5.50	2.60	1.70	7.01	4.16	3.27	1.60
Average Delay		4.30	2.10	1.22	**6.17**	**4.02**	**3.15**	**1.95**

Table 8.14 : Summary of Experimental result of S-BR-VNS - Stochastic (UK) - All (RD)

Tables 8.13 and 8.14 present a summary of all results in section 8.4.1 which showed that CPLEX was effective for solving minor and general disruption RSPs. S-BR-VNS was the best choice for the major disruption RSP and decreased the expected total train delay by 1.95% better than S-VNS, 3.15% better than S-BR-IG-LS, 4.02% better than S-IG-BR and 6.17% better than BDS-Stoch. Moreover, the reliability data showed that when the level of uncertainty increased, the reliability decreased. Moreover, when we compared the reliability between S-BR-VNS and the other methods, S-BR-VNS and S-VNS had highest reliability level at 83%, then S-IG-BR and S-BR-IG-LS also had the same reliability level at 82%. Moreover, the BDS-Stoch only had 55% of the reliability level.

Different reliability levels between all levels of disruptions were discussed with regard to how the reliability was affected by size of disruptions. We compared Tables 8.9 and 8.11which showed that the reliability of minor and general disruption was higher than the major disruption. This means that if the size of disruption increased, the reliability decreased.

8.4.2. State Railway of Thailand

Table 8.15 shows the experimental results of BDS-Stoch, S-IG-BR, S-BR-IG-LS, S-VNS and S-BR-VNS. The first two columns show the detail of each instance. After that, BDS-Stoch, S-IG-BR, S-BR-IG-LS, VNS and BR-VNS columns provided the results of total delay and reliability. Finally, at the end of the table, we calculated an average of the total delay and RD to provide a clearer understanding. All delays time are presented in **minutes** and a positive number of RD showed that the proposed method provided an improvement.

Date	Instance	BDS-Stoch		S-IG-BR		S-BR-IG-LS		S-VNS		S-BR-VNS	
		Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability	Delay	Reliability
1/2/2019	THAF101	2140.26	0.46	2132.87	0.63	1922.13	0.63	2001.34	0.68	2025.60	0.96
2/2/2019	THAF102	2222.09	0.43	2141.47	0.77	2187.60	0.83	2056.23	0.85	1941.55	0.80
3/2/2019	THAF103	2069.37	0.33	2046.06	0.91	1801.55	0.63	1767.44	0.62	1761.30	0.88
4/2/2019	THAF104	1547.01	0.26	1368.37	0.81	1249.54	0.83	1221.23	0.69	1198.00	0.68
5/2/2019	THAF105	2289.36	0.24	2284.32	0.72	2262.59	0.70	2224.50	0.89	2079.66	0.87
6/2/2019	THAF106	1665.72	0.52	1618.95	0.69	1522.68	0.83	1226.27	0.90	1195.25	0.62
7/2/2019	THAF107	2637.56	0.20	2500.98	0.88	2466.71	0.60	2442.41	0.93	2162.51	0.69
8/2/2019	THAF108	2768.12	0.68	2692.45	0.71	2561.55	0.69	2552.20	0.71	2408.17	0.82
9/2/2019	THAF109	2571.86	0.67	1984.98	0.61	1845.15	0.66	1799.37	0.80	1807.82	0.62
10/2/2019	THAF110	1643.44	0.21	1520.22	0.74	1517.03	0.74	1465.65	0.64	1411.30	0.70
11/2/2019	THAF111	1855.58	0.70	1826.68	0.67	1774.14	0.75	1793.28	0.61	1753.78	0.72
12/2/2019	THAF112	1981.48	0.53	1951.20	0.61	1884.80	0.81	1835.53	0.96	1757.61	0.81
13/2/2019	THAF113	2101.69	0.32	1807.63	0.66	1774.24	0.68	1739.56	0.73	1698.18	0.83
14/2/2019	THAF114	2117.11	0.33	1944.44	0.73	1913.95	0.67	1839.86	0.60	1835.43	0.71
15/2/2019	THAF115	2866.43	0.53	2645.20	0.68	2615.02	0.81	2571.67	0.97	2560.02	0.66
16/2/2019	THAF116	3026.04	0.55	2797.11	0.81	2763.69	0.64	2734.95	0.62	2635.65	0.95
17/2/2019	THAF117	2285.48	0.65	2273.10	0.93	2275.90	0.90	2256.65	0.74	2250.49	0.74
18/2/2019	THAF118	2583.71	0.26	2570.77	0.62	2472.62	0.64	2355.80	0.77	2205.12	0.61
19/2/2019	THAF119	1689.85	0.65	1537.25	0.75	1472.98	0.92	1441.22	0.89	1374.13	0.69
20/2/2019	THAF120	1854.07	0.60	1829.25	0.79	1769.53	0.81	1695.92	0.65	1697.55	0.77
21/2/2019	THAF121	3916.94	0.44	3212.71	0.68	3200.53	0.60	3192.23	0.93	3017.71	0.92
22/2/2019	THAF122	3993.31	0.51	3794.86	0.77	3770.80	0.84	3674.02	0.76	3661.68	0.65
23/2/2019	THAF123	3767.96	0.27	3460.03	0.65	3393.42	0.78	3337.86	0.75	3244.39	0.68
24/2/2019	THAF124	3055.96	0.48	3051.14	0.87	3024.50	0.82	3015.21	0.76	3014.11	0.71
25/2/2019	THAF125	3561.76	0.22	3400.97	0.78	3392.38	0.72	3388.26	0.65	3217.76	0.72
26/2/2019	THAF126	5145.14	0.52	5091.63	0.71	5090.09	0.83	4900.02	0.76	4861.53	0.93
27/2/2019	THAF127	3595.63	0.36	3283.33	0.90	3224.73	0.98	3150.09	0.63	3097.55	0.65
28/2/2019	THAF128	3207.96	0.53	2969.39	0.89	2976.17	0.73	2906.38	0.67	2880.51	0.96
Average Delay		2648.60	0.44	2490.62	0.75	2433.07	0.75	2378.04	0.76	2312.66	0.76

Table 8.16 : Stochastic Thai: BR-VNS Results

Date	Instance	S-VNS			S-BR-VNS			
		RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with BDS-Stoch	RD (%) compared with S-IG-BR	RD (%) compared with S-BR-IG-LS	RD (%) compared with S-VNS
1/2/2019	THAF101	6.49	6.17	-4.12	5.36	5.03	-5.38	-1.21
2/2/2019	THAF102	7.46	3.98	6.01	12.62	9.34	11.25	5.58
3/2/2019	THAF103	14.59	13.62	1.89	14.89	13.92	2.23	0.35
4/2/2019	THAF104	21.06	10.75	2.27	22.56	12.45	4.12	1.90
5/2/2019	THAF105	2.83	2.62	1.68	9.16	8.96	8.08	6.51
6/2/2019	THAF106	26.38	24.26	19.47	28.24	26.17	21.50	2.53
7/2/2019	THAF107	7.40	2.34	0.99	18.01	13.53	12.33	11.46
8/2/2019	THAF108	7.80	5.21	0.36	13.00	10.56	5.99	5.64
9/2/2019	THAF109	30.04	9.35	2.48	29.71	8.93	2.02	-0.47
10/2/2019	THAF110	10.82	3.59	3.39	14.13	7.16	6.97	3.71
11/2/2019	THAF111	3.36	1.83	-1.08	5.49	3.99	1.15	2.20
12/2/2019	THAF112	7.37	5.93	2.61	11.30	9.92	6.75	4.25
13/2/2019	THAF113	17.23	3.77	1.95	19.20	6.06	4.29	2.38
14/2/2019	THAF114	13.10	5.38	3.87	13.31	5.61	4.10	0.24
15/2/2019	THAF115	10.28	2.78	1.66	10.69	3.22	2.10	0.45
16/2/2019	THAF116	9.62	2.22	1.04	12.90	5.77	4.63	3.63
17/2/2019	THAF117	1.26	0.72	0.85	1.53	0.99	1.12	0.27
18/2/2019	THAF118	8.82	8.36	4.72	14.65	14.22	10.82	6.40
19/2/2019	THAF119	14.71	6.25	2.16	18.68	10.61	6.71	4.65
20/2/2019	THAF120	8.53	7.29	4.16	8.44	7.20	4.07	-0.10
21/2/2019	THAF121	18.50	0.64	0.26	22.96	6.07	5.71	5.47
22/2/2019	THAF122	8.00	3.18	2.57	8.30	3.51	2.89	0.34
23/2/2019	THAF123	11.41	3.53	1.64	13.90	6.23	4.39	2.80
24/2/2019	THAF124	1.33	1.18	0.31	1.37	1.21	0.34	0.04
25/2/2019	THAF125	4.87	0.37	0.12	9.66	5.39	5.15	5.03
26/2/2019	THAF126	4.76	3.76	3.73	5.51	4.52	4.49	0.79
27/2/2019	THAF127	12.39	4.06	2.31	13.85	5.66	3.94	1.67
28/2/2019	THAF128	9.40	2.12	2.34	10.21	2.99	3.21	0.89
Average Delay		10.22	4.52	2.26	**12.68**	**7.15**	**4.95**	**2.75**

Table 8.15 : Stochastic Thai: BR-VNS Results (RD)

Tables 8.15 and 8.16 show the results of stochastic RSP on the Thailand case study for expected total train delay, reliability and RD (%). In Table 8.15, S-BR-VNS and S-VNS provided highest reliability at 76% with only 1% better than S-IG-BR and S-BR-IG-LS. Moreover, the BDS-Stoch had low reliability at 44%. Then, Table 8.16 provided a fair comparison between BDS-Stoch, S-IG-BR, S-BR-IG-LS, S-VNS and S-BR-VNS. The results showed that S-BR-VNS had lower expected total train delay than the other solution methods at 2.75% better than S-VNS, 4.95% better than S-BR-IG-LS, 7.15% better than S-IG-BR and 12.68% better than BDS-Stoch. Therefore, S-BR-VNS was recommended to solve the stochastic RSP for the Thailand case study because it was best in terms of average total delay and reliability. A comparison between S-BR-VNS and S-VNS also proved that the biased randomised heuristic improved the performance of the basic VNS algorithm.

8.5. Chapter Conclusion

This chapter proposed BR-VNS with MCS techniques to provide simulated expected train delay and predicted delays. This turned the solution approach into stochastic, called S-BR-VNS. We proposed using S-BR-VNS because (i) BR-VNS improved the solution quality of BR-IG-LS for the deterministic RSP as discussed in Chapter 5 and (ii) MCS is more effective for simulating expected values.

Stochastic optimisation was based on the formulation from the previous chapter with the main objective of minimising total train delays. For S-BR-VNS procedures, MCS was added into RSP by simulating an expected value for late departure time, running time, and dwell delay into the initial solution. After that, the BR-VNS improved the rescheduling process by using the stochastic optimisation model based on the formulation from Chapter 7 with the aim to minimise expected total train delays. The MCS used the mean and variance values of test instances to generate the expected solutions.

Data for experimentation was obtained from the Southeastern train company, UK (Khosravi, 2013) and the State Railway of Thailand. For computational experimentation, two types of delay were used: (i) minor and general disruption and (ii) major disruption considered for the UK case study, while historical data was used as the test instances for the Thailand case study. Moreover, C# was used for experimentation with this problem. The average result from Chapter 7 was used for analysis and comparison.

The S-BR-VNS algorithm provided the best solution and high level of reliability among the other methods within the 400 seconds time frame. Moreover, the expected total train delays of S-BR-VNS were lower than BDS-Stoch. Therefore, we can confirm that it was beneficial to generate estimated expected total train delays which can lead to reduction in total train delays.

Chapter 9: Conclusions and future work

Nowadays, railway companies face increased challenges in train operational management with greater numbers of passengers and large and complex rail networks. Network Rail (2019), the main rail operator in the UK, reported that 37.5% of trains are late arrivals and need to be rescheduled for a smooth service. Moreover, the State Railway of Thailand (2019) uses old technology for timetable scheduling and rescheduling. Therefore, railway companies turn to operation research methodologies such as heuristics, metaheuristics and sim-optimisation methods to solve the RSP. Real-life rail network operation processes have to deal with unexpected events which occur randomly in the system and are therefore unpredictable and unavoidable. Unexpected events focused on in this thesis included running time delay, dwell time delays and late departure time. Effective railway scheduling and rescheduling are very important for dealing with these disruptions to reduce total train delay which directly relates to company expenses and quality of service. Moreover, the availability of rescheduling plans is the main responsibility of a rail company because this can reduce train delays and improve customer satisfaction.

The data for computational experimentation were sourced from two real-life case studies of

(i) Southeastern train operating company, UK that consisted of 27 trains and 135 blocks. This part of the network has complex infrastructure as bottleneck areas and dense traffic on interconnected lines for passengers in and out of London. Moreover, the UK rail network uses four-aspect signalling for rail dispatching management.

(ii) State Railway of Thailand that consisted of 101 trains and 332 blocks. Data were collected from the whole of the Thailand rail network including four main lines as Northern line, Eastern line, North-eastern line and Southern line. All trains start and end at Bangkok providing dense traffic in and out of the city. The Thailand rail network uses two-aspect signalling for rail dispatching management.

Implementation of the proposed solution methods for all case studies was conducted using CPLEX and C# on a personal computer, Intel Core i7-4500U CPU with 1.80GHz-2.4GHz, and 8GB RAM.

9.1. Research findings and contributions

This section summarises the main findings and contributions as follows:

The main findings from the literature review in Chapter 2 presented that most studies in areas of RSP focused on deterministic disruption management which the objective to minimising overall delay and total running time. However, only few researchers considered the real-life RSP with stochastic uncertainties disruption. This is a very large and complex system; therefore, to fulfil this knowledge gap we proposed using heuristics, metaheuristics and sim-optimisation to solve the RSP. We provided knowledge related to the railway process, disruption managements and the solution methods studied in the literature. We filled this knowledge gap by developing an optimisation model and implementing other solution methods such as heuristics, metaheuristics and sim-optimisation methods to integrate with state-of-the-art techniques for solving both deterministic and stochastic RSPs.

In Chapter 3, we developed a deterministic optimisation model for solving RSP by considering minimising total train delays. The main contribution of this chapter is provided a solution approach to solve the problem which called IG-BR. The IG heuristic was successfully implemented with a job shop scheduling problem and other COPs, then combine it with biased randomisation technique. The IG heuristic is simple to implement and parameter free. The biased randomisation technique was combined during the construction phase to increase the consideration of solution in a search space based on a geometric probability distribution. Performance of the biased randomisation technique was presented as a probability to rank each element in the list. The first element of the candidate list was selected as the lowest running time. To the best of our knowledge, the IG-BR algorithm has not been previously proposed to solve deterministic RSP. The limit of computation time was set at 400 seconds within the time frame of 15 minutes determined in SSDP for the update the service. The result of IG-BR presented an improvement of solution in the reduction of total train delays in two real-life case studies from the Southeastern train company, UK by 6.20% and State Railway of Thailand by 18.5%. However, when focusing on minor and general disruption instances of the UK case study, the best result was the CPLEX at 6.34% better than IG-BR.

Chapter 4 developed BR-IG-LS to improve the solution quality of IG-BR in Chapter 3. LS increased the effectiveness of the algorithm search procedure by choosing neighbourhoods through iterative improvement. Therefore, we added LS during the construction phase of the IG-BR procedure. The LS did not examine all possible solutions which resulted in decreasing computation time and increasing the size of the search space. According to the literature, our proposed methodology of using BR-IG-LS to solve the deterministic RSP by considering minimising total train delay has not been previously studied. For experimental results, the BR-IG-LS showed a decrease in total train delays compared with IG-BR from Chapter 3. Our proposed BR-IG-LS improved the solution of IG-BR by 1.3% for the UK case study and 2.09% for the Thailand case study. However, if we focused on minor and general disruption instances of the UK case study, the best result was provided by CPLEX at 4.34% better than BR-IG-LS.

The main contribution of Chapter 5 was to use the VNS algorithm combined with biased randomisation technique, called BR-VNS. VNS has been proven successful in solving train scheduling and routing; however, this has never been previously combined, with biased randomisation to solve the RSP. The BR-VNS process was applied using biased randomisation to generate an initial solution of RSP before sequencing by VNS. The methodology of VNS was performed as follows: (i) shake phase to generate random new initial conditions to avoid cycling, (ii) swap moves or insert moves to find a local optimal called local search phase, and (iii) move or not phase to check the algorithm loop. Computational experimentation of our proposed BR-VNS presented outstanding results with reduction of total train delays. When comparing BR-VNS with other algorithms from previous chapters, the BR-VNS showed the lowest result at 9.04% better than CPLEX, 3.03% better than IG-BR and 1.75% better than BR-IG-LS for all UK test instances. For the case study of the railway company in Thailand, results of BR-VNS presented a reduction of total train delays at 23.68% lower than the company historical solution, 11.14% lower than CPLEX, 6.35% lower than IG-BR and 4.35% lower than BR-IG-LS. When we considered minor and general disruption of the UK case study, the CPLEX also provided the best solution but there was a very small gap between BR-VNS and CPLEX at 1.87%. Overall, the proposed BR-VNS outperformed the other methods for large and complex disruptions and provided a good solution in limited computation time.

In Chapter 6, the main contribution was to model a stochastic optimisation of the RSP and integrate a solution methodology for dealing with disruptions to provide stability and flexibility for the railway company. The objective of the optimisation model was also to minimise total train delay for the UK and Thailand rail network characteristics. For the solution approach, a proposed S-IG-BR was developed with MCS to provide simulated expected running time, dwell time and late departure time delays. The procedure of S-IG-BR started by generating an initial solution from case studies. After that, MCS transformed the deterministic RSP into stochastic RSP by simulating expected values of running time, dwell time and late departure time delays into schedules. All predicted delay times were based on mean and variance of the test instances. The biased randomisation technique was combined in the construction phase of the IG heuristic to increase diversification of the searching procedure based on geometric distribution. This step was looped until a better solution was not found. After that, the algorithm provided the expected total train delays. Computational experimentation results of stochastic RSP with the expected train delays were compared to the deterministic RSP with no expected train delays (BDS-Stoch). Results showed that it was beneficial to generate estimated expected train delays which then led to reduction in expected total train delays at 1.95% for minor and general disruption and 2.26% for major disruption in the UK case study. Moreover, S-IG-BR provided a 5.96% improvement in the Thailand case study. Furthermore, in terms of the reliability, S-IG-BR was more robust than BDS-Stoch and provided fewer failure solutions by 27% for the UK case study and 31% for the Thailand case study. To the best of our knowledge, no other studies have proposed S-IG-BR with MCS to solve the stochastic RSP.

The main contribution of Chapter 7 was to improve the solution of S-IG-BR from the previous chapter. Therefore, S-IG-BR was combined with LS, called S-BR-IG-LS, to minimise the total train delay. To the best of our knowledge, S-BR-IG-LS has not been reported in the literature. The main reason for integrating LS with IG-BR was to increase efficiency of the LS performance and provide improvement to the IG-BR solution as discussed in Chapter 4. The procedure of S-BR-IG-LS started after an initial solution was generated, then MCS randomly simulated the expected values of running time, dwell time and late departure time delays into deterministic schedules by using mean and variance of the historical data or test instances. Then the BR-IG-LS algorithm described in Chapter 4 was used to find the expected total train delays. According to computational experimentation results, S-BR-IC-LS improved the quality of solution better than S-IG-BR from the previous chapter. In more detail, average total train delays from the S-BR-IG-LS improved the result of S-IG-BR by 1.19% for minor and general disruption and 0.88% for major disruption of the UK case study. Moreover, average total train delays of the Thailand case study between S-IG-BR and S-BR-IG-LS also improved by 2.31 %. Furthermore, the reliability of S-IG-BR and S-BR-IG-LS were equal with 82% for UK instances and 75% for Thailand instances.

The main contribution of Chapter 8 proposed the S-BR-VNS for solving the stochastic RSP to minimise expected total train delay. To satisfy the main contribution, MCS was combined with the BR-VNS algorithm already discussed in Chapter 5. MCS was used because it provided a fair comparison between the algorithms and BR-VNS and was more effective than BR-IG-LS. To the best of our knowledge, S-BR-VNS has not previously been presented in the literature to solve stochastic RSP. The procedure of S-BR-VNS started by improving the initial solution by using biased randomisation and then the MCS provided expected delays for the stochastic purposes. We followed three phases as shaking phase, local search phase and move or not phase of the basic VNS process to generate the solution of the stochastic RSP. Results from the implementation of S-BR-VNS decreased average expected total train delay in all cases of minor delays and major delays. For example, in the UK case study of a benchmark problem, S-BR-VNS reduced average total train delays at 3.15% lower than S-BR-IG-LS, 4.02% lower than S-IG-BR and 6.17% lower than the best deterministic result. In the case study of the State Railway of Thailand, the result of S-BR-VNS was 4.95% better than S-BR-IG-LS, 7.15% better than S-IG-BR and 12.68% better than the best deterministic result. Therefore, within the computation limit, average expected total train delays of S-BR-VNS outperformed the other methods. We concluded that S-BR-VNS was the best choice to solve the RSP with uncertain disruptions. In terms of reliability, S-BR-VNS was better than IG-BR and BR-IG-LS by 1% for both the UK and Thailand case studies.

9.2. Impact on the Thai railway industry

For the railway company, increase in train delay directly increases in passenger waiting and travelling time while passenger satisfaction decreases. An effective management system would improve railway operational process performance. According to the high complexity of the rail network, operational research techniques are significant methods for rail management systems. In this thesis, we proposed methodologies based on optimisation techniques such as IG with Biased Randomised, Biased Randomised IG with Local Search and Biased Randomised VNS, Sim-IG with Biased Randomised, Sim-Biased Randomised IG with Local Search and Sim-Biased Randomised VNS to present reasonable results in the computation time limit of experimentation for solving deterministic and stochastic RSP.

In Thailand, all trains are operated by the State Railway of Thailand and most of the rail network is managed by a manual system involving human-based decision-making.

Generally, the decision-makers dispatch and allocate trains following the first in first out (FCFS) rules from their own experience. This results in low performance, higher levels of train delay and waste of infrastructure usage.

An effective railway rescheduling process will decrease total train delays and improve railway management efficiency. Therefore, the proposed methodology of Biased Randomised VNS provides high performance because it shows the greatest reduction of total train delay by 23.68% compared to the current solution of the Thai railway company. Therefore, Biased Randomised VNS is suitable to deal with real-life RSP in Thailand.

According to the real-world situation, disruptions can occur anytime and lead to low passenger satisfaction, train delays and delay propagation. The railway company faces many disruptions which directly impact the railway operation process. To handle disruptions, operational decisions based on stability and flexibility to reschedule are required. According to the data from State Railway of Thailand, the sim-optimisation solution methods as Sim-IG with Biased Randomised, Sim-Biased Randomised IG with Local Search and Sim-Biased Randomised VNS were developed to increase stability and flexibility into RSP. The proposed methodology of Sim-Biased Randomised VNS is most suitable for the Thailand case study because it provides the lowest total train delay in all test instances compared to other methods. Moreover, the reliability of Biased Randomised VNS is also higher than other methods and can minimise total train delays for the RSP within the time limit required to update the service.

The railway company in Thailand does not have any software for railway management. The company should invest in railway management technology. After presenting the results of these methodologies to the State Railway of Thailand, the company noted the results of our proposed methodologies of Biased Randomised VNS and Sim-Biased Randomised VNS. They gave positive feedback that these methodologies could be effective options of their railway management process in terms of system performance. However, they also voiced concern on the budget required to implement the new railway operational system.

9.3. Limitations

Although this research completed its aims, some unavoidable limitations existed as follows:

- The proposed optimisation model only focused on headway and signalling constraints. For a more realistic solution rail transfer time, train speed and train priority should also be considered.
- This proposed optimisation model and algorithms can be classified as reactive scheduling. This only focuses on how to handle the disruption when it has already occurred.
- In the UK case study, the data for actual delay and some track information was not available because of the confidentiality issues. Therefore, it was difficult to evaluate the performance of the solution methods.
- The Thailand case study data were difficult to collect because the railway operates by using a manual process. Moreover, we waited a long time after contacting the State Railway of Thailand. The process of approval was very complex as the company is operated by the Thai Government.

9.4. Future work

This thesis could be extended as follows:

- Integration of routing and scheduling to develop a more realistic railway system management process.
- Integration of railway scheduling and crew scheduling or rolling stock management to improve an efficiency of the planning process.
- Investigate a robust optimisation model which can absorb unexpected events that occur during the train operation process for easier rescheduling and improve efficiency of the rail process.
- Examine the relationships between customer satisfaction and train delay to better understand the consideration of an acceptable delay range by passengers.
- Include more constraints such as train priority, interconnection journey and mix of passenger and freight trains.
- Our proposed algorithms could be applied to other rail networks or benchmark problems to compare performance and efficiency.
- Ongoing discussion with the Thai railway company to implement a train management system based on the results of this thesis.

References

Abid, M. M., Khan, M. B., & Muhammad, I. (2015). Timetable Optimization For Single Line Railway. *European Transport-Trasporti Europei*, (58), 1–17.

Agustin, A., Gruler, A., Armas, J. de, & Juan, A. A. (2016). Optimizing Airline Crew Scheduling Using Biased Randomization: A Case Study. In *Conference of the Spanish Association for Artificial Intelligence* (pp. 331–340). Cham: Springer. https://doi.org/10.1007/b136277

Amraoui, A. El, & Mesghouni, K. (2014). Train scheduling networks under time duration uncertainty. *IFAC Proceedings Volumes (IFAC-PapersOnline), 19*, 8762–8767.

Andersson, E. V. (2014). *Assessment of Robustness in Railway Traffic Timetables (Doctoral dissertation)*. Linköping University, Norrköping, Sweden.

Andersson, E. V, Peterson, A., & Törnquist Krasemann, J. (2013). Quantifying railway timetable robustness in critical points. *Journal of Rail Transport Planning and Management*, *3*(3), 95–110. https://doi.org/10.1016/j.jrtpm.2013.12.002

April, J., Glover, F., Kelly, J. P., & Laguna, M. (2003). PRACTICAL INTRODUCTION TO SIMULATION OPTIMIZATION. In *Proceedings of the 2003 Winter Simulation Conference* (pp. 71–78).

Arakawa, M., Fuyuki, M., & Inoue, I. (2003). An optimization-oriented method for simulation-based job shop scheduling incorporating capacity adjustment function. *International Journal of Production Economics*, *85*(3), 359–369. https://doi.org/10.1016/S0925-5273(03)00122-1

Assad, A. A. (1980). Modelling of rail networks: Toward a routing/makeup model. *Transportation Research Part B: Methodological*, *14*(1), 101–114. https://doi.org/10.1016/0191-2615(80)90036-3

Azad, N., Hassini, E., & Verma, M. (2016). Disruption risk management in railroad networks: An optimization-based methodology and a case study. *Transportation Research Part B: Methodological*, *85*, 70–88. https://doi.org/10.1016/j.trb.2016.01.001

Beckmann, M. J., McGuire, C. B., & Winsten, C. B. (1955). *Studies in the Economics of Transportation. Yale University Press, New Haven*.

https://doi.org/10.1057/jors.1980.83

Bertsimas, D., Brown, D. B. D., & Caramanis, C. (2010). Theory and Applications of Robust Optimization. *Operations Research*, 50. https://doi.org/10.1137/080734510

Borndörfer, R., & Schlechte, T. (2007). Models for Railway Track Allocation, *2007*, 62–78. Retrieved from http://vesta.informatik.rwth-aachen.de/opus/volltexte/2007/1170/pdf/07001.BorndoerferRalf.Paper.1170.pdf

Borndörfer, R., Schlechte, T., & Swarat, E. (2012). Railway track allocation - Simulation, aggregation, and optimization. *Lecture Notes in Electrical Engineering*, *148 LNEE*(September), 53–69. https://doi.org/10.1007/978-3-642-27963-8_6

Brännlund, U., Lindberg, P. O., Nõu, A., & Nilsson, J.-E. (1998). Railway Timetabling Using Lagrangian Relaxation. *Transportation Science*, *32*(4), 358–369. https://doi.org/10.1287/trsc.32.4.358

Bräysy, O. (2003). A Reactive Variable Neighborhood Search for the Vehicle-Routing Problem with Time Windows. *Informs Journal on Computing*, *15*(4), 347–368.

Bresina, J. L. (1996). Heuristic-biased stochastic sampling. *AAAI/IAAI*, *1*, 271–278.

Brucker, P., Heitmann, S., & Knust, S. (2005). Scheduling railway traffic at a construction site. In *Container Terminals and Automated Transport Systems* (pp. 345–356). Berlin, Heidelberg: Springer.

Budai-Balke, G. (2009). *Operations research models for scheduling railway infrastructure maintenance (Doctoral dissertation)*. Erasmus University Rotterdam, Rotterdam, Netherlands.

Bussieck, M. (1998). *Optimal Lines in Public Rail Transport (Doctoral dissertation)*. Technical University of Braunschweig, Braunschweig, Germany.

Bussieck, M. R., Winter, T., & Zimmermann, U. T. (1997). Discrete optimization in public rail transport. *Mathematical Programming*, *79*(1–3), 415–444. https://doi.org/10.1007/BF02614327

Cacchiani, V., & Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research*, *219*(3), 727–737. https://doi.org/10.1016/j.ejor.2011.11.003

Cáceres-Cruz, J. (2013). *Randomized Algorithms for Rich Vehicle Routing Problems : From a Specialized Approach to a Generic Methodology (Doctoral dissertation)*. Universitat Oberta de Catalunya - Internet Interdisciplinary Institute, Castelldefels, Spain.

Caimi, G. C. (2009). *Algorithmic decision support for train scheduling in a large and highly utilised railway network (Doctoral dissertation)*. ETH ZURICH, Zürich, Switzerland. https://doi.org/doi.org/10.3929/ethz-a-005947637 Rights

Caimi, G., Fuchsberger, M., Laumanns, M., & Lüthi, M. (2012). A model predictive control approach for discrete-time rescheduling in complex central railway station areas. *Computers and Operations Research*, *39*(11), 2578–2593. https://doi.org/10.1016/j.cor.2012.01.003

Caprara, A., Fischetti, M., Guida, P. L., Monaci, M., Sacco, G., & Toth, P. (2001). Solution of real-world train timetabling problems. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on* (p. 10 pp.). https://doi.org/10.1109/HICSS.2001.926325

Caprara, A., Fischetti, M., & Toth, P. (2002). Modeling and Solving the Train Timetabling Problem. *Operations Research*, *50*(5), 851–861. https://doi.org/10.1287/opre.50.5.851.362

Caprara, A., Kroon, L., Monaci, M., Peeters, M., & Toth, P. (2007). Chapter 3 Passenger Railway Optimization. *Handbooks in Operations Research and Management Science*, *14*(C), 129–187. https://doi.org/10.1016/S0927-0507(06)14003-7

Carvalho, A. N., Oliveira, F., & Scavarda, L. F. (2016). Tactical capacity planning in a real-world ETO industry case: A robust optimization approach. *International Journal of Production Economics*, *180*, 158–171. https://doi.org/10.1016/j.ijpe.2016.07.019

Chica, M., Juan Pérez, A. A., Cordon, O., & Kelton, D. (2017). Why Simheuristics? Benefits, Limitations, and Best Practices When Combining Metaheuristics with Simulation. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2919208

Chong, C. S., Sivakumar, A. I., & Gay, R. (2003). Simulation-based scheduling for dynamic discrete manufacturing. *Proceedings of the 2003 Winter Simulation Conference S.*, 1465–1473. https://doi.org/10.1109/WSC.2003.1261590

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

Corman, F., D'Ariano, A., Hansen, I. A., Pacciarelli, D., & Pranzo, M. (2011). Dispatching trains during seriously disrupted traffic situations. *2011 International Conference on Networking, Sensing and Control, ICNSC 2011*, (April), 323–328. https://doi.org/10.1109/ICNSC.2011.5874901

Corman, F., D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2010). A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological*, *44*(1), 175–192. https://doi.org/10.1016/j.trb.2009.05.004

D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2007). A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, *183*(2), 643–657. https://doi.org/10.1016/j.ejor.2006.10.034

Derbel, H., Jarboui, B., Chabchoub, H., Hanafi, S., & Mladenović, N. (2011). A variable neighborhood search for the capacitated location-routing problem. *Logistics (LOGISTIQUA), 2011 4th International Conference On*, 514–519.

Dobney, K., Baker, C. J., Quinn, A. D., & Chapman, L. (2009). Quantifying the effects of high summer temperatures due to climate change on buckling and rail related delays in south-east United Kingdom, *251*(November 2008), 245–251. https://doi.org/10.1002/met

Donzella, F., del Cacho Estil-les, M., Bersani, C., Sacile, R., & Zero, L. (2018). Train scheduling and rescheduling model based on customer satisfaction. Application to Genoa railway network. In *2018 13th Annual Conference on System of Systems Engineering (SoSE)* (pp. 593–600). IEEE. https://doi.org/10.1109/SYSOSE.2018.8428780

Driessel, R., & Mönch, L. (2011). Variable neighborhood search approaches for scheduling jobs on parallel machines with sequence-dependent setup times, precedence constraints, and ready times q. *Computers & Industrial Engineering*, *61*(2), 336–345. https://doi.org/10.1016/j.cie.2010.07.001

Fischetti, M., Salvagnin, D., & Zanette, A. (2009). Fast Approaches to Improve the Robustness of a Railway Timetable. *Transportation Science*, *43*(3), 321–335. https://doi.org/10.1287/trsc.1090.0264

Frantzén, M., Ng, A. H. C., & Moore, P. (2011). A simulation-based scheduling system for real-time optimization and decision making support. *Robotics and Computer-Integrated Manufacturing*, *27*(4), 696–705. https://doi.org/10.1016/j.rcim.2010.12.006

Fügenschuh, A., Homfeld, H., Huck, A., Martin, A., & Yuan, Z. (2008). Scheduling Locomotives and Car Transfers in Freight Transport. *Transportation Science*, *42*(4), 478–491. https://doi.org/10.1287/trsc.1080.0248

Gafarov, E. R., Dolgui, A., & Lazarev, A. A. (2015). Two-station single-track railway scheduling problem with trains of equal speed. *COMPUTERS & INDUSTRIAL ENGINEERING*, *85*, 260–267. https://doi.org/10.1016/j.cie.2015.03.014

Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of Metaheuristics*. (F. Glover & G. A. Kochenberger, Eds.) (Second Edi). New York: Springer.

Glover, F., Kelly, J. P., & Laguna, M. (1996). New advances and applications of combining simulation and optimization. In *Proceedings of the 1996 Winter Simulation Conference* (pp. 144–152). https://doi.org/10.1016/j.ijhydene.2017.01.222

Glover, F., Kelly, J. P., & Laguna, M. (1999). NEW ADVANCES FOR WEDDING OPTIMIZATION AND SIMULATION. In *Proceedings of the I999 Winter Simulation Conference* (pp. 255–260).

Gulati, N., & Bandyopadhyay, M. (2016). An Improvement in Optimally Rescheduled Trains for Railway Traffic Network Using Randomized Linear Particle Swarm Optimization : RLPSO. *International Journal of Engineering Science and Computing*, *6*(4), 3807–3810. https://doi.org/10.4010/2016.883

Hansen, P., & Mladenovi, N. (2001). Variable neighborhood search : Principles and applications c. *European Journal of Operational Research*, *130*, 449–467.

Harish, K., Sibi, R., & Kumar, S. V. (2014). Efficient Parallel Iterated Greedy Algorithm for Solving Task Assignment Problem in Heterogeneous Systems, 47–51.

Harrod, S. (2006). Optimal Scheduling of Mixed Speed Trains on a Single Track Line.

Investment Management and Financial Innovations, *5*(4), 7–24. https://doi.org/10.1287/xxxx.0000.0000

Hassan, G. M., & Reynolds, M. (2018). Genetic Algorithms for Scheduling and Optimization of Ore Train Networks. *EPiC Series in Computing*, *55*, 81–92. https://doi.org/10.29007/fzrq

Hassannayebi, E., Sajedinejad, A., & Mardani, S. (2014). Urban rail transit planning using a two-stage simulation-based optimization approach. *Simulation Modelling Practice and Theory*, *49*, 151–166. https://doi.org/10.1016/j.simpat.2014.09.004

Hofman, M. A., & Madsen, L. F. (2005). *Robustness in train scheduling (Doctoral dissertation, Master thesis)*. IMM, DTU.

Huisman, D., Kroon, L. G., Lentink, R. M., Vromans, M. J. C. M., Huisman, D., Kroon, L. G., … Vromans, M. J. C. M. (2005). Operations research in passenger railway transportation. *Statistica Neerlandica*, *59*(4), 467–497. https://doi.org/10.1111/j.1467-9574.2005.00303.x

Huo, J., Wu, J., Kang, L., & Bo, W. (2016). Railway Timetable Rescheduling Based on Priority and Train Order Entropy. *Journal of Computing in Civil Engineering*, *30*(5). https://doi.org/10.1061/(A10.1061/(ASCE)CP .1943-5487.0000559.SCE)CP

Jespersen-Groth, J., Potthof, D., Clausen, J., Huisman, D., Kroon, L., Maróti, G., & Nielsen, M. N. (2009). *Disruption Management in Passenger Railway Transportation*. *Robust and online large-scale optimization*. Berlin, Heidelberg: Springer.

Johnson, D. S., Papadimitriou, C. H., & Yannakakis, M. (1988). How Easy Is Local Search ? *JOURNAL OF COMPUTER AND SYSTEM SCIENCES*, *37*, 79–100. https://doi.org/doi.org/10.1016/0022-0000(88)90046-3

Jotikasthira, O. (2018). Rail service on track to crisis. *29 APR 2018*. Bangkok Post (Spectrum). Retrieved from https://www.bangkokpost.com/thailand/special-reports/1454593/rail-service-on-track-to-crisis

Juan, A. A., Barrios, B. B., Vallada, E., Riera, D., & Jorba, J. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, *46*, 101–117.

https://doi.org/10.1016/j.simpat.2014.02.005

Juan, A. A., Faulin, J., Ferrer, A., Lourenço, H. R., & Barrios, B. (2013). MIRHA: Multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems. *Top*, *21*(1), 109–132. https://doi.org/10.1007/s11750-011-0245-1

Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, *2*, 62–72. https://doi.org/10.1016/j.orp.2015.03.001

Juan, A. A., Grasman, S. E., Caceres-cruz, J., & Bektas, T. (2014). A simheuristic algorithm for the Single-Period Stochastic Inventory-Routing Problem with stock-outs. *Simulation Modelling Practice and Theory*, *46*, 40–52. https://doi.org/10.1016/j.simpat.2013.11.008

Juan, A. A., Pascual, I. .-0022000088900463-main. pd., Guimarans, D., & Barrios, B. (2015). Combining biased randomization with iterated local search for solving the multidepot vehicle routing problem. *International Transactions in Operational Research*, *22*(4), 647–667. https://doi.org/10.1111/itor.12101

Kang, Q., He, H., & Wei, J. (2013). An effective iterated greedy algorithm for reliability-oriented task allocation in distributed computing systems. *Journal of Parallel and Distributed Computing*, *73*(8), 1106–1115. https://doi.org/10.1016/j.jpdc.2013.03.008

Keita, K., Pellegrini, P., & Rodriguez, J. (2016). Solving the real-time Railway Traffic Management Problem with Benders decomposition. In *ROADEF 2016, 17ème conférence dela Société française de Recherche Opérationnelle et d'Aide à la Décision*.

Khadilkar, H. (2018). A Scalable Reinforcement Learning Algorithm for Scheduling Railway Lines. *IEEE Transactions on Intelligent Transportation Systems*, (1), 1–11.

Khan, M. B., & Zhou, X. (2010). Stochastic Optimization Model and Solution Algorithm for Robust Double-Track Train-Timetabling Problem.pdf. *IEEE Transactions on Intelligent Transportation Systems*, *11*(1), 81–89.

Khosravi, B. (2013). *Train Scheduling with Application to the UK Rail Network (Doctoral* *dissertation)*. University of Southampton, Southampton, United Kingdom.

Khosravi, B., Bennell, J. J. A., Potts, C. N. C., Delling, D., & Liberti, L. (2012). Train Scheduling and Rescheduling in the UK with a Modified Shifting Bottleneck Procedure. *Atmos*, *25*, 120–131. https://doi.org/10.4230/OASIcs.ATMOS.2012.120

Kinder, M. (2008). Models for Periodic Timetabling. *Master Thesis*, 112. Retrieved from papers2://publication/uuid/E4B49923-F083-4B8F-A0F5-8E5A66F73800

Klemmt, A., Horn, S., Weigert, G., & Wolter, K.-J. rgenrgen. (2009). Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems Andreas. *Robotics and Computer-Integrated Manufacturing*, *25*, 917–925. https://doi.org/10.1016/j.rcim.2009.04.012

Korytkowski, P., Wiśniewski, T., & Rymaszewski, S. (2013). An evolutionary simulation-based optimization approach for dispatching scheduling. *Simulation Modelling Practice and Theory*, *35*, 69–85. https://doi.org/10.1016/j.simpat.2013.03.006

Kraft, E. R. (1987). A Branch and Bound Procedure for Optimal Train Dispatching. *Journal of the Transportation Research Forum*, *28*(1), 263–276.

Kroon, L., Maróti, G., Helmrich, M. R., Vromans, M., & Dekker, R. (2008). Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, *42*(6), 553–570. https://doi.org/10.1016/j.trb.2007.11.002

Kroon, L. G., Lentink, R. M., & Schrijver, a. (2008). Shunting of Passenger Train Units: An Integrated Approach. *Transportation Science*, *42*(December 2014), 436–449. https://doi.org/10.1287/trsc.1080.0243

Lambrechts, O., Demeulemeester, E., & Herroelen, W. (2008). A tabu search procedure for developing robust predictive project schedules. *International Journal of Production Economics*, *111*(2), 493–508. https://doi.org/10.1016/j.ijpe.2007.02.003

Laporte, G., & Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. In *North-Holland Mathematics Studies* (Vol. 132, pp. 147–184). Amsterdam, Netherlands: North-Holland.

Li, S., De Schutter, B., Yang, L., & Gao, Z. (2016). Robust Model Predictive Control for Train Regulation in Underground Railway Transportation. *IEEE Transactions on Control Systems Technology*, *24*(3), 1075–1083.

https://doi.org/10.1109/TCST.2015.2480839

Liao, C., & Cheng, C. (2007). A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date q. *Computers & Industrial Engineering*, *52*, 404–413. https://doi.org/10.1016/j.cie.2007.01.004

Liebchen, C., Lubbecke, M., Mohring, R., & Stiller, S. (2009). The concept of recoverable robustness, linear programming recovery, and railway applications. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5868 LNCS*, 1–27. https://doi.org/10.1007/978-3-642-05465-5_1

Liu, R., Li, S., & Yang, L. (2018). Collaborative optimization for metro train scheduling and train connections combined with passenger flow control strategy. *Omega*, (October). https://doi.org/10.1016/j.omega.2018.10.020

Liu, S. Q., & Kozan, E. (2009). Scheduling trains as a blocking parallel-machine job shop scheduling problem. *Computers and Operations Research*, *36*(10), 2840–2852. https://doi.org/10.1016/j.cor.2008.12.012

Lusby, R. M., Larsen, J., Ehrgott, M., & Ryan, D. (2011). Railway track allocation: Models and methods. *OR Spectrum*, *33*(4), 843–883. https://doi.org/10.1007/s00291-009-0189-0

Mascis, A., & Pacciarelli, D. (2002). Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, *143*(3), 498–517. https://doi.org/10.1016/S0377-2217(01)00338-1

Meng, L., & Zhou, X. (2011). Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. *Transportation Research Part B: Methodological*, *45*(7), 1080–1102. https://doi.org/10.1016/j.trb.2011.05.001

National Rail Enquiries. (2013). National Rail Enquiries - Official source for UK train times and timetables. Retrieved April 29, 2013, from http://www.nationalrail.co.uk/

Network Rail. (2019). Punctuality on the national rail network – Network Rail. Retrieved February 12, 2019, from https://www.networkrail.co.uk/who-we-are/how-we-work/performance/railway-performance/punctuality-on-the-national-rail-network/

Nicoară, E. S., Filip, F. G., & Paraschiv, N. (2011). Simulation-based optimization using genetic algorithms for multi-objective flexible JSSP. *Studies in Informatics and Control*, *20*(4), 333–344. https://doi.org/10.24846/v20i4y201101

Nielsen, L. K., Kroon, L., & Maróti, G. (2012). A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, *220*(2), 496–509. https://doi.org/10.1016/j.ejor.2012.01.037

Nitisiri, K., Gen, M., & Ohwada, H. (2019). A parallel multi-objective genetic algorithm with learning based mutation for railway scheduling. *Computers & Industrial Engineering*, *130*, 381–394. https://doi.org/10.1016/j.cie.2019.02.035

Orlin, J. B., Punnen, A. P., & Schulz, A. S. (2003). Approximate Local Search in Combinatorial Optimization. *Ssrn*, (July 2003). https://doi.org/10.2139/ssrn.423560

Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, *12*(4), 417–431. https://doi.org/10.1007/s10951-008-0090-8

Pachl, J. (2009). *Railway Operation and Control* (Fourth Edi). Washington: VTD Rail.

Parkes, D. C., & Ungar, L. H. (2001). An auction-based method for decentralized train scheduling. In *Proceedings of the fifth international conference on Autonomous agents - AGENTS '01* (pp. 43–50). New York, New York, USA: ACM Press. https://doi.org/10.1145/375735.375866

Pellegrini, P., Marlière, G., & Rodriguez, J. (2014). Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *TRANSPORTATION RESEARCH PART B*, *59*, 58–80. https://doi.org/10.1016/j.trb.2013.10.013

Pinedo, M., & Singer, M. (1999). A Shifting Bottleneck Heuristic for Minimizing the TotalWeighted Tardiness in Job Shops. *Naval Research Logistics*, *46*(1), 1–17.

Ping, L., Limin, J., & Xiong, K. (2001). Study on GA-based Train Dispatching. In *Proceedings of the 5th world congress on railway research* (pp. 949–953).

Polacek, M., Hartl, R. F., & Doerner, K. (2005). A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with Time Windows. *Journal of Heuristics*, *10*, 613–627.

Potts, C. N., & Strusevich, V. A. (2009). Fifty years of scheduling : a survey of milestones. *Journal of the Operational Research Society*, *60*, 41–68. https://doi.org/10.1057/jors.2009.2

Ribas, I., Companys, R., & Tort-Martorell, X. (2011). An iterated greedy algorithm for the flowshop scheduling problem with blocking. *Omega*, *39*(3), 293–301. https://doi.org/10.1016/j.omega.2010.07.007

Rodriguez, J. (2007). A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B: Methodological*, *41*(2), 231–245. https://doi.org/10.1016/j.trb.2006.02.006

Roshanaei, V., Naderi, B., Jolai, F., & Khalili, M. (2009). A variable neighborhood search for job shop scheduling with set-up times to minimize makespan. *Future Generation Computer Systems*, *25*(6), 654–661. https://doi.org/10.1016/j.future.2009.01.004

Rossetti, M. A. (2002). Potential Impacts of Climate Change on Railroads. *The Potential Impacts of Climate Change on Transportation.*, 1–13. https://doi.org/10.3402/gha.v4i0.7913

Rossetti, M. A. (2007). Analysis of Weather Events on U.S Railroads. In *23rd Conference on IIPS* (Vol. 4).

Roy, B., & Sussmann, B. (1964). *Les problemes d'ordonnancement avec contraintes disjonctives.pdf*. *SEMA* (Vol. 9). Paris.

Rudan, J., Kersbergen, B., Van den Boom, T., & Hangos, K. (2019). Performance analysis of MILP based model predictive control algorithms for dynamic railway scheduling, (1).

Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, *177*(3), 2033–2049. https://doi.org/10.1016/j.ejor.2005.12.009

Ruiz, R., & Stützle, T. (2008). An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, *187*(3), 1143–1159. https://doi.org/10.1016/j.ejor.2006.07.029

Şahin, İ. (1999). Railway traffic control and train scheduling based oninter-train conflict management. *Transportation Research Part B: Methodological*, *33*(7), 511–534. https://doi.org/10.1016/S0191-2615(99)00004-1

Sajedinejad, A., Mardani, S., Hasannayebi, E., Mohammadi K., S. A. R. M., & Kabirian, A. (2011). SIMARAIL: Simulation based optimization software for scheduling railway network. In *Proceedings of the 2011 Winter Simulation Conference* (pp. 3730–3741). IEEE. https://doi.org/10.1109/WSC.2011.6148066

Salido, M. A., Barber, F., & Ingolotti, L. (2008). Robustness in Railway Transportation Scheduling. *In Intelligent Control and Automation*, 2880–2885.

Samà, M., D′Ariano, A., Corman, F., & Pacciarelli, D. (2017). A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations. *Computers and Operations Research*, *78*, 480–499. https://doi.org/10.1016/j.cor.2016.02.008

Schlechte, T. (2011a). *Railway Track Allocation: Models and Algorithms*. Technischen Universität Berlin, Berlin, Germany.

Schlechte, T. (2011b). *Railway Track Allocation - Simulation and Optimization* (Vol. 32). Berlin-Dahlem, Germany.

Schlechte, T., & Borndorfer, R. (2009). *Balancing Efficiency and Robustness – A Bi-criteria Optimization Approach to Railway Track Allocation* (Vol. 22). Berlin-Dahlem, Germany.

Serafini, P., & Ukovich, W. (1989). A Mathematical Model for Periodic Scheduling Problems. *SIAM Journal on Discrete Mathematics*. https://doi.org/10.1137/0402049

Shafia, M. A., Pourseyed Aghaee, M., Sadjadi, S. J., & Jamili, A. (2012). Robust train timetabling problem: Mathematical model and branch and bound algorithm. *IEEE Transactions on Intelligent Transportation Systems*, *13*(1), 307–317. https://doi.org/10.1109/TITS.2011.2169961

Southwell, C. E. (2012). *A robust model for disaster risk and decision analysis*. *ProQuest Dissertations and Theses*. Retrieved from https://pallas2.tcl.sc.edu/login?url=http://search.proquest.com/docview/1027934863?accountid=13965%5Cnhttp://www.tdnet.com/uscc/resolver/default.asp?genre=article&issn=&volume=&issue=&title=A+robust+model+for+disaster+risk+and+decision+an

alysis&spage=&date=

State Railway of Thailand. (2019). Timetable & Fare. Retrieved February 15, 2019, from http://www.railway.co.th/main/timetable/time-train.html#

Szpigel, B. (1973). Optimal train scheduling on a single line railway. *Journal of Operation Research*, *72*, 344–351.

Takeuchi, Y., & Tomii, N. (2005). Robustness Indices for Train Rescheduling. In *1st International Conference on Railway Operations Modelling and Analysis RailDelft2005* (pp. 1–19).

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. New Jersey: John Wiley & Sons.

Tamannaei, M., Saffarzadeh, M., Jamili, A., & Seyedabrishami, S. (2016). A Novel Train Rescheduling Approach In Double-Track Railways: Optimization Model and Solution Method Based on Simulated Annealing Algorithm Mohammad. *International Journal of Civil Engineering*, *14*(3), 139–150.

Thailand Trains. (2019). Thailand Train Map _ Thailand Trains. Retrieved February 15, 2019, from http://www.thailandtrains.com/thailand-train-map/

The Railway Technical Website. (2018). Signalling | The Railway Technical Website | PRC Rail Consulting Ltd. Retrieved February 6, 2019, from http://www.railway-technical.com/signalling/

Tian, X., & Niu, H. (2019). A bi-objective model with sequential search algorithm for optimizing network-wide train timetables. *Computers & Industrial Engineering*, *127*, 1259–1272. https://doi.org/10.1016/j.cie.2018.03.012

Tornquist, J. (2006). *Railway Traffic Disturbance (Doctoral dissertation)*. Blekinge Institute of Technology, Karlskrona, Sweden. https://doi.org/10.1016/j.tra.2006.05.003

Törnquist, J., & Persson, J. a. (2005). Train Traffic Deviation Handling Using Tabu Search and Simulated Annealing. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, *00*(C), 1–10. https://doi.org/10.1109/HICSS.2005.641

Törnquist, J., & Persson, J. A. (2007). N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B: Methodological*, *41*(3), 342–362. https://doi.org/10.1016/j.trb.2006.06.001

Ushida, K., Makino, S., & Tomii, N. (2011). Increasing Robustness of Dense Timetables by Visualization of Train Traffic Record Data and Monte Carlo Simulation. In *9th World Congress on Railway Research* (pp. 1–12).

Van de Vonder, S. (2006). *Proactive-reactive procedures for robust project scheduling (Doctoral dissertation)*. KATHOLIEKE UNIVERSITEIT LEUVEN, Leuven, Belgium.

Van De Vonder, S., Demeulemeester, E., & Herroelen, W. (2007). A classification of predictive-reactive project scheduling procedures. *Journal of Scheduling*, *10*(3), 195–207. https://doi.org/10.1007/s10951-007-0011-2

Veelenturf, L. P., Potthoff, D., Huisman, D., Kroon, L. G., & Wagelmans, A. P. M. (2012). *A recoverable robust solution approach for real-time railway crew rescheduling*. Rotterdam, Netherlands.

Vromans, M. J. C. M., Dekker, R., & Kroon, L. G. (2006). Reliability and heterogeneity of railway services. *European Journal of Operational Research*, *172*(2), 647–665. https://doi.org/10.1016/j.ejor.2004.10.010

Wang, Y., Tang, T., Ning, B., van den Boom, T. J. J., & De Schutter, B. (2015). Passenger-demands-oriented train scheduling for an urban rail transit network. *Transportation Research Part C*, *60*, 1–23. Retrieved from http://pub.deschutter.info/abs/15_038.html

Wilson, M. (2016). *Robust scheduling in an uncertain environment*. Delft University of Technology. https://doi.org/10.4233/uuid:9bf41aed-ebd0-4038-8961-9fdbb699b1a7

Xia, Y., Van Ommeren, J. N., Rietveld, P., & Verhagen, W. (2013). Railway infrastructure disturbances and train operator performance: The role of weather. *Transportation Research Part D: Transport and Environment*, *18*(1), 97–102. https://doi.org/10.1016/j.trd.2012.09.008

Yang, T., Kuo, Y., & Chang, I. (2004). Tabu-search simulation optimization approach for flow-shop scheduling with multiple processors - A case study. *International Journal of Production Research*, *42*(19), 4015–4030.

https://doi.org/10.1080/00207540410001699381

Yang, X., Chen, A., Ning, B., & Tang, T. (2016). A stochastic model for the integrated optimization on metro timetable and speed profile with uncertain train mass. *Transportation Research Part B: Methodological*, *91*, 424–445. https://doi.org/10.1016/j.trb.2016.06.006

Yin, J., Tang, T., Yang, L., Gao, Z., & Ran, B. (2016). Energy-efficient metro train rescheduling with uncertain time-variant passenger demands: An approximate dynamic programming approach. *Transportation Research Part B: Methodological*, *91*, 178–210. https://doi.org/10.1016/j.trb.2016.05.009

Ying, K. C. (2008). Solving non-permutation flowshop scheduling problems by an effective iterated greedy heuristic. *International Journal of Advanced Manufacturing Technology*, *38*(3–4), 348–354. https://doi.org/10.1007/s00170-007-1104-y

Ying, K. C., & Cheng, H. M. (2010). Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic. *Expert Systems with Applications*, *37*(4), 2848–2852. https://doi.org/10.1016/j.eswa.2009.09.006

Yuan, Z., Armin, F., Homfeld, H., & Balaprakash, P. (2008). Iterated Greedy Algorithms for a Real-World Cyclic Train Scheduling Problem, (October). https://doi.org/10.1007/978-3-540-88439-2_8

Zandieh, M., & Adibi, M. A. (2010). Dynamic job shop scheduling using variable neighbourhood search. *International Journal of Production Research*, *48*(8), 2449–2458. https://doi.org/10.1080/00207540802662896

Zhang, C., Gao, Y., Yang, L., Kumar, U., & Gao, Z. (2019). Integrated optimization of train scheduling and maintenance planning on high-speed railway corridors. *Omega*, *87*, 86–104. https://doi.org/10.1016/j.omega.2018.08.005