UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING AND THE ENVIRONMENT

Civil, Maritime and Environmental Engineering and Science Academic Unit

Improving Pre-Trip Information about Transfer-Involved Rail Routes: Algorithms and Analytical Methods

by

Yiwei Guo

Thesis for the degree of Doctor of Philosophy

May 2018

[historical version: June 2017]

Abstract

Small delays and major disruptions are frequently encountered in rail passenger transport, which brings challenges not only to railway timetabling and operations but also to timetable-based passenger information. This thesis is aimed at identifying the unresolved problem(s) in the existing pre-trip timetable information systems and at developing a set of novel algorithms and analytical models to enhance the pre-trip timetable information about and the understanding of those transfer-involved routes within a national-level railway network. Specifically, it tries to answer the following four inter-related questions: i) which transfer-involved routes are the weaknesses in terms of pre-trip timetable information, among the numerous origin-destination pairs; ii) how to develop an effective and easy-to-implement approach to coping with these weaknesses; iii) how to quantify and know in advance the potential effect of a specific information improvement strategy; and iv) what are the potential factors that render some of the transfer-involved routes particularly vulnerable to delays and disruptions.

Since the research touches on multiple disciplines, the relevant concepts in railway timetabling and operations, journey planning algorithms, statistical analysis, and decision theory are firstly introduced. Built on these fundamentals and an introduction to the concepts of critical transfers and critical routes, a screening algorithm is developed that is able to efficiently identify those transfer-involved rail routes that may be particularly vulnerable to delays and disruptions and may need information enhancements. After that, by reviewing the pros and cons of existing methods, a novel historical-data-driven algorithm is developed to deal with those weaknesses in terms of pre-trip timetable information. In order to obtain a more precise estimation of the potential effect of a particular information enhancement strategy, an analytical framework is developed that is able to evaluate a specific strategy ex ante. The underlying assumptions are presented and the potential limitations are discussed. All of the algorithms and models presented in this thesis have been extensively tested by exploiting the open data from British railways, the results of which are promising in terms of efficiency and effectiveness. Some interesting findings are presented about British railways, followed by a discussion of potential directions in future research.

Preface

The limits of my language mean the limits of my world.
– Ludwig Wittgenstein

Born into a traditional Chinese family, I have been deeply influenced since my childhood by family values and traditional East Asian philosophies such as Ruism, Daoism, and Buddhism. A big thankyou firstly goes to my parents, Yong and Yuxian. Both of them are decent ordinary people in China who have been contributing a lot to the society and to the family. Due to historical reasons, they were unable to get a good education when they were young, which has been the biggest regret in their lives. Therefore, they have been hoping that their only child could become a well-educated man and make a difference in his life. This thesis could not have been smoothly generated without my parents' unconditional support. They will continue to be the role models for me, and their diligence, tenacity, and modesty will continue to guide me through the ups and downs for the rest of my life.

Apart from my parents, special thanks also go to my supervisor, Professor J.M. Preston. The completion of this thesis would be impossible without his continuous encouragement and guidance along the way. Being a rigorous academic and an old-school English gentleman, he possesses almost all the good qualities one can expect of a Western intellectual elite: self-discipline, diligence, pursuit of excellence, humour, etc. A Chinese proverb says, 'Teachers are your second parents (一日为师, 终生为父)'. Indeed, I have learned a lot from Professor Preston over the past four years, not only about how to be a qualified student but also about how to be a respectable person.

Moreover, the quality of this thesis would have been discounted without the constructive comments and sincere advice from a number of other academic staff in Transportation Research Group: Dr Ben Waterson, Prof Tom Cherrett, Dr John Armstrong, Dr Simon Box, Dr Simon Blainey, to name a few. Here, I want to express my immense gratitude to all of them.

Last but not least, I would like to express my most profound thanks to all those great minds in history for motivating and inspiring me during my study. Einstein said, 'Although I am a typical loner in daily life, my consciousness of belonging to the invisible community of those who strive for truth, beauty, and justice has preserved me from feeling isolated'. My own experience during the four-year study largely coincides with this quote (although I am not a fan of Einstein and am such an obscure novice researcher): there have been difficult times during the process of the four-year study, but those invisible great spirits have granted to me a lot of strength each time I felt frustrated and was about to give up.

'A journey of a thousand miles begins with a single step (千里之行, 始于足下)'. If the past four years study at Southampton can be defined as a single step (neither large nor small), this step largely prepares me for a long journey ahead, whether the journey is composed of smooth curves or sharp zigzags.

Yiwei Guo
June 2017
Southampton, UK

Contents

List of Figures

List of Tables

List of Algorithms

Glossary

Note: Only those frequently encountered terminology and symbols are listed here.

Terminology

AW: is short for actual window, which represents the time window between the actual (/recorded) arrival of a feeder train and the actual departure of the corresponding connecting train.

Critical Transfer: an introduced concept which can be roughly described as a highly risky (in terms of probability and consequence) transfer plan that is (to be) recommended (by a journey planning system).

Critical Itinerary: an extension of the concept of critical transfer, corresponding to a whole journey between a given pair of origin and destination stations.

Critical Route: involves at least one (generic) transfer, defined mainly in the context of pre-trip information about recommended itineraries. The list of recommended itineraries (generated by a journey planning system) for a critical route is characterised by a high percentage of critical itineraries.

CRF: is short for Critical Routes Finder, which is the developed screening algorithm (presented in Chapter 3) for efficiently locating those critical routes within a given railway network.

CSA: is short for Connection Scan Algorithm, which forms the core of the self-developed journey planning simulator in this thesis.

GTFS: is short for General Transit Feed Specification, which is a popular data format for exchanging transit information. The timetable data (i.e. the National Rail timetable data) adopted in this thesis is in GTFS format.

IPS: is short for Itinerary-oriented Performance Statistics, which can be viewed as the itinerary-level version of PPM (public performance measure). Moreover, IPS is also the name of a proposed algorithm in Chapter 4 (c.f. Algorithm 3 in Section 4.3).

LAT: is short for latest-tolerable arrival time, which is a concept that is introduced in Section 5.4 to serve as a boundary condition for determining whether the disutility of a particular passenger has been increased.

MTT: is short for minimum transfer time, also called 'connection time' or 'minimum connection time' in the literature, representing the length of time that must elapse between the advertised arrival time of a feeder train and the advertised departure time of the connecting train within a railway station. That is, the connection between two trains is officially valid only if it satisfies the constraint of the corresponding 'minimum connection time'. In realistic railway timetabling and operations, MTT is often station-specific and is a heuristic measurement (a rough estimation) of a 'typical' connection within a particular railway station.

MUI: is short for most uncertain interval, which is network-specific and can be determined by analysing the big data about arrival delays within a given railway network.

NTT: is short for net transfer time, which can be viewed as a connection-specific MTT. NTT is coined to emphasise the difference in granularity between MTT and NTT.

PBPM: is short for Performance-Based Pre-Modification of advertised arrival times, which is the core algorithm proposed in Chapter 4. The involved technicalities can be found in Chapters 4 and 5.

PBPM+: is an augmented version of PBPM, which could generate alternative itineraries when necessary.

RPM: is short for Route-oriented Performance Measure, which can be viewed as an extension of PPM and IPS. RPM not only can be used to evaluate route-specific punctuality and reliability, but also can be used to quantify the impact of modified pre-trip information.

RUM: is short for Route-oriented Utility Measure, which is devised to incorporate more realistic factors in evaluating the effectiveness of the proposed algorithmic solution to critical routes.

SW: is short for scheduled window, representing the time window between the scheduled arrival of a feeder train and the scheduled departure of its corresponding connecting train.

TAL: is short for threshold for arrival lateness, which is an absolute standard that is adopted for determining whether an 'average' passenger choosing a particular itinerary has arrived at his/her destination station on time (based on the advertised arrival time).

Abbreviations

ATOC: is short for Association of Train Operating Companies, being a body that represents the 23 train operating companies that provide passenger services on the privatised British railway system. ATOC has been renamed Rail Delivery Group since October 2016.

BSB: denotes Bournemouth – Southampton Central – Brighton, one of the studied routes in Chapter 5.

ECB: denotes Ebbw Vale Town – Cardiff Central – Birmingham New Street, one of the studied routes in Chapter 5.

HMN: denotes Harwich Town – Manningtree – Norwich, one of the studied routes in Chapter 5.

ILM: denotes Ilkley – Leeds – Middlesbrough, one of the studied routes in Chapter 5.

KWN: denotes Knottingley – Wakefield Kirkgate – Nottingham, one of the studied routes in Chapter 5.

KYS: denotes London Kings Cross – York – Scarborough, one of the studied routes in Chapter 5.

LMD: denotes Liverpool Lime Street – Manchester Piccadilly – Doncaster, one of the studied routes in Chapter 5.

NRE: is short for National Rail Enquiries, which is the official source of customer information for all passenger rail services in Great Britain (excluding some of the urban rail services within Greater London).

P1: denotes observation Period 1 in the evaluations conducted in Chapter 5, corresponding to the relevant data records between 12 October 2015 and 4 December 2015.

P2: denotes observation Period 2 in the evaluations conducted in Chapter 5, corresponding to the relevant data records between 25 January 2016 and 18 March 2016.

P3: denotes observation Period 3 in the evaluations conducted in Chapter 5, corresponding to the relevant data records between 13 June 2016 and 5 August 2016.

P4: denotes observation Period 4 in the evaluations conducted in Chapter 5, corresponding to the relevant data records between 3 October 2016 and 25 November 2016.

P5: denotes observation Period 5 in the evaluations conducted in Chapter 5, corresponding to the relevant data records between 16 January 2017 and 10 March 2017.

PPM: is short for public performance measure, which is the industry standard of British railways for measuring the punctuality and reliability of train services.

RBH: denotes Rugeley Trent Valley – Birmingham New Street – Hereford, one of the studied routes in Chapter 5.

RIL: is short for recommended itinerary list (c.f. Sections 4.3 and 4.4).

RTT: is short for Realtime Trains, which is the source of those historical train movements data adopted in this thesis.

RVT: is short for Route-View Timetable, which is the major data structure underlying the evaluations and analyses in Chapters 4 and 5.

SML: denotes Sudbury (Suffolk) – Marks Tey – London Liverpool Street, one of the studied routes in Chapter 5.

TOC: is short for train operating companies, representing those private rail operators of passenger routes within Britain's railway network.

Symbols

$arr_s(\cdot)$: represents the scheduled arrival time of a particular train service at a particular railway station. The two variants of this symbol – arr_s_XX and $sch_{r,p,j,k}$ – have the same meaning with it. (c.f. Section 4.3, Section 4.5, Section 5.2, and Section 5.4)

$arr_m(\cdot)$: represents the pre-modified (advertised) arrival time of a particular train service at a particular railway station. The two variants of this symbol – arr_m_XX and $md_{r,p,j,k}$ – have the same meaning with it. (c.f. Section 4.3, Section 4.5, Section 5.2, and Section 5.4)

$arr_a(\cdot)$: represents the actual/reconstructed arrival time of a particular train/itinerary at a particular railway station. The two variants of this symbol – arr_a_XX and $act_{r,p,j,k}$ – have the same meaning with it. (c.f. Section 4.3, Section 4.5, Section 5.2, and Section 5.4)

$dep_s(\cdot)$: represents the scheduled departure time of a particular train/itinerary from a particular railway station. The variant of this symbol – dep_s_XX – has the same meaning with it. (c.f. Section 4.3, Section 4.5, Section 5.2)

$E(\cdot)$: represents the expected value (i.e. average/mean value) of a given variable/statistic (c.f. Section 5.3 and Section 5.5).

$jt_0(\cdot)$: represents the average journey time under the scenario in which there are no missed transfers (c.f. Section 4.3).

$jt_1(\cdot)$: represents the average journey time under the scenario in which there is exactly one missed transfer (c.f. Section 4.3).

$jt_m(\cdot)$: represents the pre-modified (advertised) journey time of a recommended itinerary (c.f. Section 4.3).

$\delta(\cdot)$: represents the average delay of the connecting train (of a recommended itinerary) at the destination station (c.f. Section 4.3).

RPM_s: represents the calculated RPM for a given observation period assuming that the unmodified pre-trip information is adopted about scheduled arrival times (c.f. Section 5.3).

RPM_p: represents the obtained RPM for a given observation period assuming that the modified pre-trip information has been adopted about pre-modified arrival times (generated by the PBPM algorithm) (c.f. Section 5.3).

ΔRPM: $= RPM_p - RPM_s$, representing the change (in RPM) the modified pre-trip information could have brought (c.f. Section 5.3).

t_θ : is involved in Section 3.5, can be interpreted as the 'latest tolerable arrive time' of a feeder train, beyond which the corresponding transfer would be missed. Note that t_θ should not be confused with the concept of LAT in Section 5.4. t_θ is train-oriented, while LAT is passenger-oriented.

T_j : is involved in the proposed algorithms in Chapter 4, representing a particular train service in a series of involved train services in a recommended itinerary.

Chapter 1
Introduction

1.1 Motivation

Rail transport has a long history and is accessible to the public in most countries across the world. As a traditional sector and a natural monopoly, the rail industry has inevitably built itself an image of a relatively closed system lacking efficiency and often follows quite different development models in different countries. Rail transport in Britain, like that in many other European countries (e.g. Sweden, France, the Netherlands, etc.), takes a model of vertical separation of train operators and infrastructure managers to increase on-track competition (Kurosaki, 2008; Mizutani et al., 2014). Although this development path could to some degree improve on openness and cost efficiency, it increases the complexity of a railway system and brings increased difficulty in reconciling the various stakeholders (i.e. passengers, train operators, infrastructure managers, public authority, and the general public) (Kurosaki, 2008; Martin, 2014).

One of the challenges currently faced by British railway and other intensely utilised European railways (e.g. Dutch and Swiss railways) is the prevalence of small delays as well as major disruptions (Figures 1.1 and 1.2). On the one hand, the rail demand is steadily increasing and the capacity utilisation is reaching its limit at critical parts (Network Rail, 2016a), which renders the rail network sensitive to delays and disruptions (i.e. the impact of a delay/disruption caused by some endogenous/exogenous factor could easily be spread across a large dispatching area). On the other hand, an extensive upgrading/renewal of rail infrastructure is expected to be a time-consuming process, following the current development path. In such a context, rail researchers in European countries (e.g. Denmark and the Netherlands) have been looking for, over the past decade or so, software solutions (e.g. advanced timetabling techniques that take into account robustness and stability, optimisation models for capacity utilisation at bottlenecks, etc.) to the problem of delay and disruption management.

Figure 1.1 Train delays Example One (Source: metro.co.uk, 12 Feb 2016)

Figure 1.2 Train delays Example Two (photo shot on 28 Nov 2016, at Southampton Central Railway Station)

In reality, however, theoretically optimal plans/schemes could not always be fully implemented due to various technical or political limitations. Take the timetabling process for example. Railway timetabling is a complicated process that needs to balance between many factors (e.g. easy-to-remember departure times at major stations, speed limits at different block sections, recovery times along long-distance routes, buffer times between conflicting train paths, etc.) and involves the collaboration between different train operators and between train operators and the infrastructure manager (Kroon et al., 2014; Network Rail, 2016a). Moreover, even if a theoretically optimal plan/scheme could be fully implemented, a globally optimal solution could not guarantee local optimality (Goverde, 2014). That is, train delays cannot be thoroughly eradicated in a large railway system, for there will always be certain elements of the various operational processes that could not be fully optimised (Yuan and Medeossi, 2014).

Observing that an extensive upgrading of rail infrastructure is almost unlikely to happen in the foreseeable future and that existing operator-oriented software solutions have their own limitations, this thesis tries to tackle the problem of delay and disruption management from a different angle and tries to provide a passenger-oriented software solution to deal with those blind spots over which current technologies have little control. A catalyst for generating the idea of adopting a passenger-oriented methodology is the so-called journey planning systems (Figure 1.3) that have been gaining popularity in the developed world over the last decade or so.

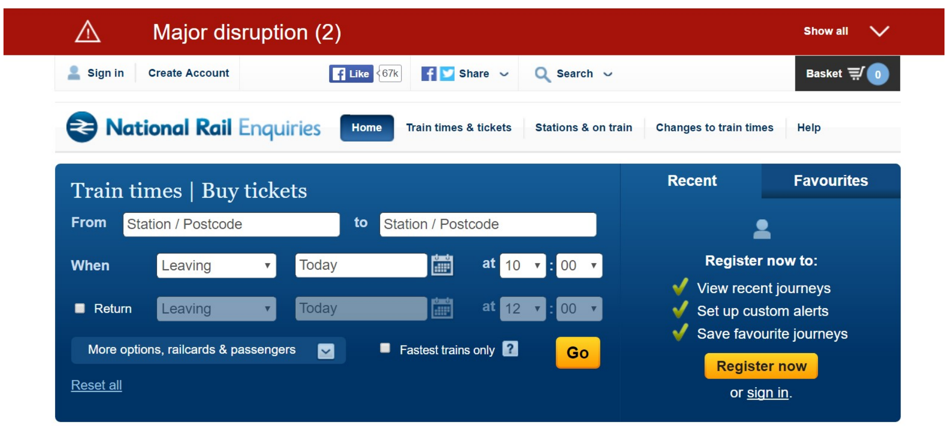


Figure 1.3 National Rail Enquiries – an example of journey planning systems (Source: www.nationalrail.co.uk, accessed 29 Dec 2016)

As an important interface between passengers and train operators, a journey planning system (e.g. National Rail Enquiries in Britain) usually offers a wide range of online services related to rail travel: from timetable-based itinerary planning to live disruption alerts, and from online ticketing to promotional information. The core functionality of a journey planning system is undoubtedly the itinerary planning part, for live disruption information and ticketing services can also be obtained later at railway stations. The demand for computer-aided itinerary planning is especially significant for those long-distance and/or unfamiliar journeys (Farag and Lyons, 2008), and such a journey often involves one or more transfer activities en route. Due to the periodicity of the railway timetable, those recommended itineraries (journey plans) are often cyclic and hence can be grouped by route. Compared with direct rail routes (lines), those long-distance, transfer-involved rail routes are more prone to delays and disruptions due to the additional risk of missed transfers. However, current technologies (i.e. algorithms behind those journey planning systems) have little control over the quality of the generated results (i.e. those recommended itineraries). The recommendations are derived from the underlying (planned) timetables, the quality of which is further dependent upon the timetabling techniques adopted. Unfortunately, due to the aforementioned reasons, no such perfect timetable design exists in reality that could absorb all perturbations in a railway network and is resistant to major disruptions in the network. Therefore, those long-distance and transfer-involved rail routes become a potential problem: journeys following such routes often need to be pre-planned with the aid of journey planning systems, but the quality of the pre-trip information about these journeys is often disregarded (i.e. the actual journey times and arrival times often significantly exceed their advertised counterparts).

Table 1.1 Rail journeys in Britain: by purpose and frequency (Source: DfT, 2013)

Percentages

	Commuting	Business	Leisure	Total
5 or more days a week	77	12	8	52
2-4 days a week	17	14	11	15
Once a week	3	10	11	6
1-3 times a month	2	18	18	8
Less than once a month	1	23	28	10
First time have made this journey	1	23	24	9
Total	**100**	**100**	**100**	**100**

Although rail transport can be categorised as a minority mode (Preston, 2015; DfT, 2016a) and rail passengers choosing long-distance and transfer-involved routes are theoretically a minority group in Britain (DfT, 2016a; DfT, 2016b), these journeys cannot be ignored, for they are more likely to be infrequent non-commuting journeys and hence tend to be more elastic to the quality of rail services. Table 1.1 above provides a more tangible illustration of how British rail journeys are distributed on the dimensions of journey purpose and journey frequency, in which we can see that most non-commuting journeys are infrequent. Since punctuality/reliability has always been among passengers' top concerns about rail services (Bates et al., 2001; ATOC, 2013; DfT, 2016b) and passengers' experience of punctuality/reliability is increasingly dependent on the quality of information provided before and during delays/disruptions (Lyons et al., 2008; Ben-Elia et al., 2013; van der Hurk, 2015), the potential gains (losses) from improving (disregarding) the quality of information about this particular group of long-distance and transfer-involved journeys will be non-negligible in the long run, considering the overall magnitude of daily rail travel (DfT, 2016b).

1.2 Research question and objectives

The main research question/problem of this thesis is as follows.

How to exploit train movements data (train operation records) to develop an efficient and effective methodology for practical use to improve the pre-trip information about those critical transfer-involved routes within a national-level intercity rail (passenger) transport system, taking into account not only the constraint of capacity utilisation but also the interplay between the competitiveness (/attractiveness) of and the reliability (/punctuality/robustness) of the recommended itineraries?

To answer this question, it is essential to have an in-depth understanding of the characteristics of the accessible train movements data (mainly about Britain's passenger rail system in this thesis), to study the state-of-the-art pre-trip information systems/prototypes/algorithms and identify the gap between the existing solutions/ideas and a reasonably good solution to the research problem, and to develop an effective and practicable solution based on a comprehensive grasp of the relevant issues and concepts from a variety of disciplines and prove its advantages over the existing ones through quantitative and/or qualitative analyses.

More specifically, this research comprises the following four objectives.

1) Formulate the problem of pre-trip timetable information about those transfer-involved routes, and identify those weak points within the existing pre-trip information systems.

2) Review the existing algorithmic solutions/ideas to tackle missed transfers, and identify the inadequacies of the existing methodologies and knowledge.

3) Develop an effective and easy-to-implement solution to the research problem, and develop an analytical framework that is able to quantify the quality (potential effect) of a given information enhancement strategy.

4) Collect, analyse, and exploit real-world train movements data, and evaluate the developed solution approaches in terms of efficiency and effectiveness.

1.3 Multi-disciplinary research

The main body of this thesis involves/blends the concepts and methods from a number of different disciplines, and a thorough understanding of these fragmented but inter-related pieces of knowledge is vital to the understanding of the algorithms and models developed and presented in this thesis. More specifically, this thesis touches mainly on the following fields:

- Algorithm Engineering: the design and implementation of the algorithms in this thesis cannot be achieved without an in-depth understanding of the algorithmic-level mechanisms of current journey planning systems, or without a mastery of the various programming techniques.

- Probability and Statistics: the information enhancing algorithm and the analytical framework are historical-data-driven and involve statistical analyses. Knowing about the principles and underlying assumptions of Statistics may facilitate the understanding of the technicalities of the relevant models and algorithms.

- Mathematical Optimisation: although not directly involved, the understanding of the optimisation techniques behind railway timetabling and journey planning is necessary for the understanding of this thesis.

- Railway Engineering: good knowledge of rail-related devices and daily operational practices could help better understand the screening algorithm and the statistical analyses.

- Data Science: the screening algorithm and the analytical framework involve massive data processing, which requires advanced programming skills to control the computational complexity.

- Decision Theory: the analytical framework introduces a series of assumptions on passengers' choices, and it is necessary to have a good understanding of the basics of Decision Theory.

In order to facilitate the understanding of the main body of this thesis, a concentrated introduction to the fundamentals of the relevant disciplines is to be presented in Chapter 2.

Engineering problems are often quite complicated. In the remainder of this thesis, the reader may find it full of technicalities and pieces of terminology borrowed from different disciplines, which renders it not that readable. This is, however, not surprising – a relatively straightforward idea does not mean an equally simple implementation in reality. Cross-disciplinary cooperation is not as easy as imagined – the trend of persistent specialisation[1] seems to be pushing professionals of different fields away from each other. In fact, previous studies (e.g. Porter and Rafols, 2009) have shown that inter-disciplinary cooperation nowadays is largely limited to neighbouring fields. Therefore, policy makers in the rail sector should think about how to design a sustainable mechanism to truly strengthen inter-disciplinary cooperation between the various departments of the rail sector.

1.4 Thesis structure

The main body of this thesis is composed of seven chapters. Following this general introduction in Chapter 1, Chapter 2 presents a concentrated introduction to the fundamentals of the relevant disciplines.

Chapter 3 formulates the problem of pre-trip timetable information about those transfer-involved routes by introducing the concepts of Critical Transfers and Critical Routes, and

[1] http://undsci.berkeley.edu/article/modern_science

presents details about an efficient screening algorithm that has been developed (and tested on British timetable data) to identify those Critical Routes within a given railway system.

Chapter 4 contains a review of the existing algorithmic approaches to enhancing pre-trip timetable information, and details a historical-data-based algorithmic approach that is tailored to tackling Critical Routes. Real-world examples in British railways (e.g. the route Bournemouth → Southampton → Brighton) are also presented in this chapter to illustrate why the proposed approach is more able (compared with the existing algorithmic approaches) to deal with Critical Routes.

Chapter 5 describes an analytical framework that is specially developed to quantify the impact of information enhancement strategies and conduct ex-ante analysis of those identified Critical Routes. The underlying assumptions are systematically sorted out, followed by a small fictitious example that is employed to illustrate the intermediate calculations. After that, the results and their implications are presented from a number of case studies of the identified Critical Routes in Britain's passenger rail system. A detailed introduction to the data utilised and the considerations in parameter selection are firstly presented, followed by the obtained results and the key findings from these case studies.

Chapter 6 concludes this thesis and recommends directions for further research.

Chapter 2
Fundamentals

2.1 Introduction

This chapter is intended to provide a concentrated introduction to the fundamentals of several different disciplines, from which the algorithms and models presented in subsequent chapters are designed and developed. Instead of ambitiously pursuing rigor and comprehensiveness, the writing of this chapter strives for clarity and conciseness. Unlike those encyclopaedic textbooks, this chapter tries to deliver just enough information about the most relevant concepts to this thesis.

Sections 2.2 and 2.3 belong to the domain of Algorithm Engineering, in which the introduced concepts are closely related to the algorithms presented in Chapters 3 and 4. Section 2.4 explains several key concepts in Probability and Statistics, the applications of which can be found in Chapters 3 – 5. Section 2.5 introduces the fundamentals of Railway Timetabling and Operations, which are essential to the understanding of Chapters 3 – 5. Section 2.6 elucidates several important concepts in Decision Theory, and they are mainly touched on in Chapter 5. Section 2.7 summarises this chapter.

2.2 Graph Theory and Shortest-paths Problems

This section introduces the relevant concepts in Graph Theory and Shortest-paths Problems, which are the theoretical foundation for the various Journey Planning Algorithms (to be introduced in the next section). The latter (i.e. the various Journey Planning Algorithms) form the core of current journey planning systems. For a more detailed picture of Graph Theory and Shortest-paths Problems, it is recommended to refer to Cormen et al. (2009) and Diestel (2010).

2.2.1 Graphs

In Mathematics, a *graph* is defined as a 2-tuple (V, E) in which V represents a set of *vertices* (or *nodes*) and E is a collection of *edges* (or *arcs*) defining on set V the pairwise relationships between its member vertices. Two vertices u and v from set V are said to be *adjacent* if an edge e = (u, v) exists in E, and the two vertices are called *end vertices* of edge e. Also, we say a vertex v and an edge e is *incident* with each other if v is an end vertex of e. Two edges from collection E are called *parallel edges* if they have the same end vertices, and an edge e from E is called *self-loop* if its two end vertices are the same. Two non-parallel edges from collection E are called *adjacent* if they share a common end vertex. An edge e = (u, v) from collection E is said to be *directed* if the pair is ordered (u preceding v), otherwise it is *undirected*.

We say a graph G = (V, E) is a *directed graph* (or *digraph*) if all edges in E are directed. Likewise, a graph with all its edges being undirected is called an *undirected graph*. A *path* p in a graph G is a sequence of adjacent edges $<e_1, e_2, \ldots, e_k>$ where e_i belongs to E for all i in the range [1, k]. When there are no parallel edges in G, a path P can also be represented as a sequence of adjacent vertices $<v_1, v_2, \ldots, v_k, v_{k+1}>$ where v_i is the source vertex of e_i for all i in the range [1, k] and v_{k+1} is the target vertex of e_k. A graph is said to be *weighted* if each of its edges is associated with a *weight*, given by a *weight function* w: E → R (R represents the set of real numbers).

2.2.2 Single-source shortest-paths problem

Given a weighted digraph G = (V, E), the *path weight* of a path p in G is defined as the summation of the weights of its component edges. And a *shortest path* from a source vertex u to a target vertex v in G is any feasible path from u to v that has the minimum path weight. An important application of graph theory in transportation is a problem set called *shortest-paths problems*, all of which aim to find shortest paths between certain pairs of vertices in a graph. Generally speaking, there are three categories of shortest-paths problems: *single-source shortest-paths problem*, *single-pair shortest-path problem*, and *all-pairs shortest-paths problem*. For different categories of shortest-paths problems, different types of algorithms can be applied to solve them.

Given a graph G, the *single-source shortest-paths problem* aims to find a shortest path from a certain source vertex s to every vertex reachable from s in G. This category is like a 'baseline' of all shortest-paths problems. To solve this category of shortestpaths problems, the solutions are different for different kinds of graphs. Given an unweighted graph G = (V, E) in which all of its edges have unit weights, the *breadthfirst-search* algorithm can be applied to solve this problem. It can be proven that breadth-first search is a linear-time algorithm with a time complexity of O(|V| + |E|) if the unweighted graph is implemented using adjacency lists.

2.2.3 Dijkstra's algorithm

Given a weighted digraph G = (V, E) in which all of its edge weights are nonnegative, *Dijkstra's algorithm* can work on it to efficiently solve the single-source shortest-paths problem. Since in the graph representation of transportation networks edge weights satisfy this nonnegative restriction, Dijkstra's algorithm is frequently used to solve routing problems in transportation networks. Dijkstra's algorithm can be classified as a *label setting algorithm*, which is characterised by scanning each vertex at most once in the execution of the algorithm. The runtime of Dijkstra's is in O(|E| + |V|log|V|) if its constituent priority queue is implemented using a Fibonacci heap, and this bound can be further improved in some cases using well-designed data structures.

2.2.4 Bellman-Ford algorithm

If there are negative edge weights in a given weighted digraph G = (V, E), Dijkstra's algorithm will no longer guarantee the correctness of the routing results. In this case, *Bellman-Ford algorithm* can be used to complete the task. Bellman-Ford is a *label correcting algorithm*, which means that each vertex may be scanned several times during an execution. The worst-case running time of Bellman-Ford is O(|V||E|), slower than Dijkstra's, but can become competitive with Dijkstra's in certain scenarios.

2.2.5 Other members of the shortest-paths family

The symmetric problem of single-source shortest-paths problem is the so-called *single-destination shortest-paths problem*, which can be solved as a single-source shortest-paths problem by simply reversing the direction of each edge.

The *single-pair shortest-path problem* is aimed at finding a shortest path from a given source vertex s to a given target vertex t within a given graph G. This category of shortest-paths problems can be conveniently solved by applying one of the abovementioned algorithms designed for single-source shortest-paths problems, and the running time can be at least as fast as that of the counterpart in the single-source scenario.

To find a shortest path from one vertex to another for each pair of vertices in a graph is the goal of the *all-pairs shortest-paths problem*. For a given graph G = (V, E), this category of shortest-paths problems can be solved either by repeatedly calling one of the algorithms designed for single-source shortest-paths problems, or by applying algorithms tailored for this category. The *Floyd-Warshall algorithm* is designed for solving all-pairs shortest-paths problems, and the runtime of this algorithm is in $\Theta(|V|^3)$. It can be proven that for dense graphs, Floyd-Warshall runs faster than |V| calls to Dijkstra's. *Johnson's algorithm* is another tailored algorithm for all-pairs shortest-paths problems, and for sparse graphs this algorithm is asymptotically faster than both Floyd-Warshall and repeated execution of Dijkstra's.

2.3 Journey Planning Problems and Algorithms

The various journey planning systems currently in use are driven by a family of mathematical models and algorithms called Journey Planning (or Route Planning) Problems and Algorithms, which forms an emerging branch of Algorithm Engineering that has been developing over the last decade or so. Although scientific knowledge about these algorithms remains rather fragmented, their applications (e.g. Google Maps and the various traveller information media) have been gaining popularity around the world due to the boom of Information and Communications Technologies (ICT). This section tries to extract from the large body of literature in this area the most relevant information about current journey planning technologies, and re-organise these pieces of knowledge in an easy-to-understand way.

2.3.1 The evolution of journey planning

Traditionally, people plan routes/journeys manually using some kind of printed 'travel guides'. These travel guides can take the form of roadmaps for car drivers or timetables for public transport riders. Routing manually can be an enjoyable experience if the traveller is time-and-cost-insensitive and just enjoys driving or riding. In other cases, this can be a time-consuming process that relies heavily on the traveller's past experience and the quantity and quality of travel information stored on those guides. Nowadays, with the development of information technology and computing techniques, traditional travel information carriers such as road maps, timetables, etc. can be digitally stored and integrated. The task of routing/journeying can therefore be efficiently performed on these digitalised transportation networks by computers equipped with well-designed *journey planning algorithms*, the core component of journey planning systems.

According to Wagner (2015), the history of route planning algorithms in transportation networks can be categorised into five phases: I) Theoretical explorations (1959 – 1999); II) The emergence of speed-up techniques (1999 – 2005); III) The applications in road networks (2005 – 2008); IV) Towards more realistic scenarios in car & public transport (2008 – 2012); and V) New challenges on customisability, multimodality, etc. (since 2012).

2.3.2 Modelling transport networks as graphs: road vs. rail

If we model intersections as vertices and road segments as weighted edges, it would be convenient to convert a road map into a weighted digraph and hence one of the above-mentioned shortest-paths algorithms could be applied on the converted digraph to compute shortest paths with respect to some chosen criterion (e.g. travel time). In rail networks (and other timetable-based public transport networks), in contrast, the application of graph theory and shortest-paths algorithms is not that straightforward. This is due to an important difference between road and rail networks: in most cases, road segments can be traversed at any time during a day, whereas track segments can only be traversed at discrete time points. In other words, timetables (corresponding to rail networks) often contain additional temporal information than roadmaps (corresponding to road networks) and this additional information needs to be taken into account when converting a timetable into a graph.

Abstractly, a *timetable* can be viewed as a 4-tuple (S, Z, C, D), where S is a set of *stations*, Z is a set of *trains*, C is a set of *elementary connections*, and D is a set of *service dates*. In this 4-tuple, an elementary connection in set C is itself a 5-tuple (Z_i, S_d, S_a, t_d, t_a), which can be interpreted as follows: a train Z_i departs the current stop station S_d at time t_d and arrives at the immediately next stop station S_a at time t_a. At a given station S_j, a rail passenger can *transfer* from one train to another if and only if the *time window* between the arrival of the feeder train and the departure of the connecting train is no less than a predefined station-specific *minimum transfer time* $\tau(S_j)$.

2.3.3 Time-Expanded Model vs. Time-Dependent Model

Basically, there are two types of graph models in the literature to represent a timetable: *time-expanded model* and *time-dependent model*. While both of them are well-studied in the literature, neither of them can be said a perfect representation of a timetable. In practice, they have their respective application areas and meanwhile they have their own limitations.

The time-expanded model builds an *event graph* (or *time-expanded graph*) for a given timetable to 'unroll' time (Bast et al., 2015). In the basic version, departure events and arrival events are modelled as vertices and the relationships between events are modelled as edges (Schulz et al., 2000). In the realistic version, additional transfer vertices are introduced to incorporate minimum transfer times (Müller-Hannemann and Schnee, 2007; Pyrga et al., 2004; Pyrga et al., 2008). An advantage of a time-expanded model lies in its flexibility and robustness in the application in *multi-criteria optimisation* (which will be explained later in more detail). A disadvantage of a time-expanded model is that the converted graph for a timetable is usually very large and hence consumes more storage space than other models such as the time-dependent model (Pyrga et al., 2004). In order to overcome this disadvantage, several techniques have been devised to compress the resulting graph (for more details about these techniques, please refer to Delling et al. (2009) and Pyrga et al. (2008)).

Unlike time-expanded model, the *time-dependent model* does not create a vertex for each departure and arrival event but represents stations and/or routes as vertices and utilises complex edge weights to model timetable information. In the basic version, vertices represent stations and edges are associated with *travel time functions* to 'encode' departure and arrival times (Brodal and Jacob, 2004). In the realistic version, apart from station vertices, additional route vertices are created to take into account minimum transfer times (Pyrga et al., 2008). The number of route vertices in the realistic version can be reduced by merging disconnected route vertices at a station into one (more details about this technique can be found in Delling et al. (2012)). The advantage of the time-dependent model is its low memory consumption compared with its time-expanded counterpart, but time-dependent graphs have a limitation that it is not easy to apply speedup techniques on a time-dependent framework in some real-world applications due to the cumbersome edge weights (Berger et al., 2009). Figures 2.1 – 2.3 give an illustration of how a timetable can be converted into a realistic time-expanded or time-dependent graph.

train 1		
stations	times	
A	dep	9:30
B	arr	10:03
	dep	10:05
C	arr	11:00

train 2		
stations	times	
B	dep	10:10
D	arr	10:50

train 3		
stations	times	
B	dep	10:15
E	arr	10:50

train 4		
stations	times	
B	dep	10:40
D	arr	11:20

Figure 2.1 An imaginary rail system consisting of only four train services (in which ‘service date’ can be thought of as ‘everyday’)

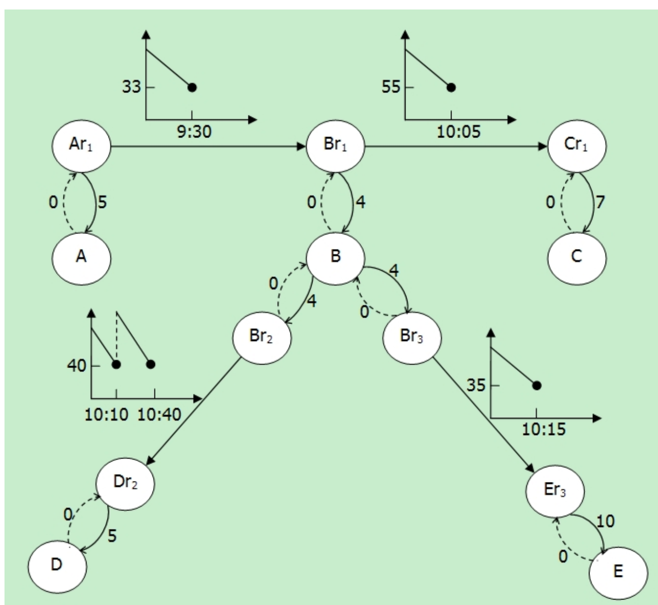


Figure 2.2 The time-dependent graph constructed from the timetables in Figure 2.1 (there are five station vertices A, B, C, D, and E, and the other seven vertices are route vertices corresponding to the three routes A to C via B, B to D, and B to E. The two edges between a station vertex and a route vertex are transfer edges in which the solid edge is assigned a station-specific minimum transfer time for checking, and the dotted edge has no assigned weight and is only used to link vertices. The edge between two route vertices is a route edge with which a travel time function is associated. A travel time function maps, for each train traversing this edge, the departure time to the travel time.)

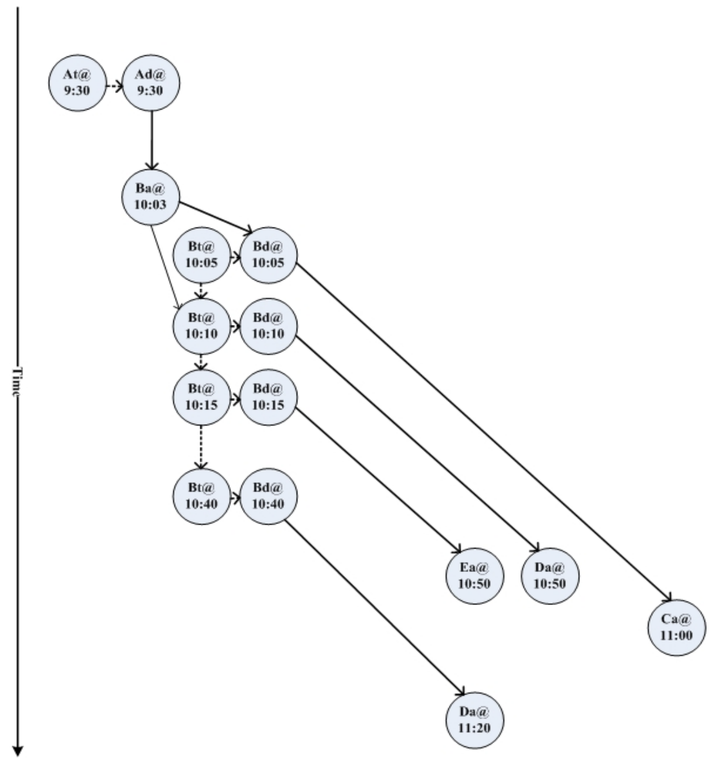


Figure 2.3 The time-expanded graph built from the timetables in Figure 2.1 [there are three kinds of vertices: departure vertices (e.g. Ad@9:30), arrival vertices (e.g. Ba@10:03), and transfer vertices (e.g. Bt@10:05). Each of these vertices corresponds to a specific event in the rail system. The edges between vertices have no weights and are only used for linking. Each departure vertex is associated with a transfer vertex with the same timestamp. While an arrival vertex can be directly linked to a departure vertex of the same train, it has to be linked to a transfer vertex with the restriction of a predefined minimum transfer time if a transfer (between two trains) happens.]

It should be noted that Bast and Storandt (2014) have recently proposed a new graph model called *frequency-based model*, which builds on the time-dependent framework and exploits the periodicity of public transport systems to compress the resulting graph. This model can be viewed as a relatively independent category (Bast et al., 2015).

2.3.4 Array-based models

Although it is natural and convenient to model road (car) networks as graphs, graph models (time-expanded/time-dependent) are computationally expensive in dealing with rail (timetable) networks, especially when the information provider has limited computing resources. In order to reduce computational complexity, a series of *array-based models* have been developed over the past few years, which act as alternatives to graph models. Representative algorithms that adopt array-based models are *CSA* (Connection Scan Algorithm) and *RAPTOR* (Round-bAsed Public Transit Optimized Router), both of which explicitly exploit the characteristics of public transport systems and convert expensive graph searching into operations on simple arrays. Further details about CSA and RAPTOR can be found in Dibbelt et al. (2013) and Delling et al. (2014b).

2.3.5 Earliest Arrival Problem and its solutions

With graph models and array-based models at hand, various journey planning problems in timetable networks can be effectively solved by applying the variants of shortest-paths algorithms on these models. Journey planning in timetable networks has three problem variants: *earliest arrival problem*, *range problem*, and *multi-criteria problem*.

The *earliest arrival problem* can be regarded as a benchmark. It can be roughly described as follows: given a query (s, t, τ) in which s is the source station, t is the target station, and τ is the planned departure time, how to find a *journey* (i.e. a sequence of trips and footpaths in chronological order) that departs from s no earlier than τ and arrives at t as early as possible. A query like this is often called a *time query* or *earliest arrival query*. For time queries, travel time is the only optimisation criterion considered. And since travel times between two stations are inherently nonnegative, the aforementioned Dijkstra's algorithm can be conveniently applied on a converted graph to answer this kind of queries.

When adopting the time-expanded approach, the corresponding algorithm is called *time-expanded Dijkstra (TED)*. Likewise, *time-dependent Dijkstra (TDD)* refers to the underlying graph model is of the time-dependent form. Although the application of Dijkstra's algorithm on time-expanded graphs is straightforward, the application on time-dependent graphs needs some augmentation and additional requirements on edge weights (i.e. nonnegative and FIFO (first in, first out)) should be satisfied (Orda and Rom, 1990; Orda and Rom, 1991). Based on the observation that time-expanded graphs are inherently *DAGs* (Directed Acyclic Graphs), the *Topological Sort algorithm* (see e.g. Cormen et al. (2009) for more details) can be applied to answer queries in linear time (its time complexity $\Theta(|V| + |E|)$ is faster than Dijkstra's algorithm) (Mellouli and Suhl, 2006). A larger speedup can be achieved by adopting some array-based algorithms such as CSA (Dibbelt et al., 2013).

2.3.6 Range Problem and its solutions

The *range problem* can be described as follows: given a query (s, t, [τ_1, τ_2]) in which s is the source station, t is the target station, and [τ_1, τ_2] is the range of planned departure times, how to find a set of journeys with minimum travel times that departs from s within the given time interval. A query of this form is often called a *profile query*. Variants of Dijkstra's algorithm can be applied on a time-dependent graph converted from a given timetable to solve this type of problems (cf. Dean (1999), Delling et al. (2012), and Nachtigall (1995)). The frequency-based model (Bast and Storandt, 2014) and CSA (Dibbelt et al., 2013) can also be extended to solve range problems.

2.3.7 Multi-Criteria Problem and its solutions

Unlike the earliest arrival problem and the range problem, the *multi-criteria problem* considers additional optimisation criteria (e.g. number of transfers, monetary cost, etc.) besides travel time. Given a query (s, t, τ) in which s and t are source and target stations and τ is the planned departure time, the multi-criteria problem asks for a *Pareto set* of mutually *non-dominated* journeys in terms of the chosen optimisation criteria. We say a journey J_1 *dominates* another journey J_2 if and only if J_1 is better with respect to at least one criterion and no worse with respect to the other criteria.

Although early studies (e.g. Hansen (1979)) have shown that a Pareto set can contain exponentially many results even when only two optimisation criteria are considered, the number of solutions in a Pareto set is often much smaller in real-world public transport journey planning due to the fact that there are often correlations between different optimisation criteria (Bast et al., 2015; Dibbelt et al., 2013; MüllerHannemann and Weihe, 2001). For example, *Layered Dijkstra algorithm* can be applied on a time-dependent timetable graph to convert a bicriteria optimisation (i.e. travel time and number of transfers) into a single-criterion (i.e. travel time) optimisation, which exploits the correlation between the two optimisation criteria: travel time and number of transfers (Brodal and Jacob, 2004; Pyrga et al., 2008).

For multi-criteria problems with additional optimisation criteria (apart from travel time and number of transfers), *Multi-criteria Label-Setting* (MLS) algorithms (cf. Demeyer (2013), Disser et al. (2008), and Müller-Hannemann and Schnee (2007)) or *MultiLabel-Correcting* (MLC) algorithms (cf. Dean (1999) and Delling and Wagner (2009)) can be applied on a converted timetable graph to solve them. Apart from MLS and MLC, other model-specific algorithms can also be applied to solve the multi-criteria problem. Bast and Storandt (2014) extend their query algorithm to incorporate number of transfers as an additional optimisation criterion by adopting the proposed frequency-based model. Moreover, the basic version of RAPTOR includes travel time and number of transfers as optimisation criteria, and additional criteria can be added by adopting the multi-criteria version (McRAPTOR) (Delling et al., 2014b).

2.4 Related concepts in Probability and Statistics

Probability and Statistics are two interrelated disciplines: the former places more emphasis on theory while the latter assigns more weight to applications. Although these two fields have long been regarded as an essential part of modern science, there remain important controversies within them (de Elia and Laprise, 2005; Hájek, 2012). This section is not aimed at providing a comprehensive introduction to these two highly developed fields, but tries to focus on an introduction to one of the popular theories in Probability and Statistics that is adopted in this thesis.

2.4.1 Classical probability

The concept of *probability* is one of the essential tools of statistics, which can be traced back to the 17^{th} century in the studies of games of chance. Throwing a dice, tossing a coin, and drawing a card are examples of *games of chance*, which are characterised by an uncertain outcome in a trial.

Although the outcome of each particular trial is uncertain, it is recognised that there exists a predictable long-term outcome. For example, in a large number of trials of tossing an ideal (i.e. well-balanced and symmetrical) coin, heads will turn up in about one half of these trials. This estimation/prediction of the percentage of heads in a number of trials/experiments can be obtained *a priori* before these trials have been actually conducted: since only one of the two outcomes (a head or a tail) can be obtained in a single toss of a coin, and since the coin is unbiased (symmetrical and well-balanced), equal chances would be expected of obtaining a head and obtaining a tail. The above reasoning can be formally recapitulated by the following *classical definition of probability* (Mood, 1974):

If there are n possible outcomes resulting from a random experiment and these n outcomes are mutually exclusive and equally likely, and if n_A of the n possible outcomes have an attribute A, then the probability of A is the fraction n_A/n.

In this definition, the key words are ‘mutually exclusive’, ‘equally likely’, and ‘random’. Although these conditions can be satisfied in such games/experiments as throwing a dice, tossing a coin, and drawing a card, they are not applicable in many other situations. The classical definition of probability tends to be unable to deal with an infinite number of possible outcomes (e.g. what is the probability that a randomly chosen integer be even?) and those scenarios in which the concepts of ‘symmetry’, ‘equally likely’, etc., are not applicable (e.g. what is the probability that a child born in the United Kingdom will be a girl?). Since these scenarios are frequently encountered in reality but their results cannot be obtained purely by deductive reasoning, the classical definition of probability needs to be extended to accommodate them.

2.4.2 Relative frequency

One solution to the limitations of classical probability is the so-called *relative frequency* (or *empirical probability*), which is based on actual observations.

The restrictions exerted on classical probability such as equipossible outcomes and symmetry are relaxed, and hence relative frequencies can no longer be determined a prior before the experiments have been actually conducted. That is, relative frequencies can only be obtained *a posterior* after empirical evidence has been collected. For example, a tossed coin is no longer viewed as absolutely symmetrical and balanced from the perspective of relative frequency: there always exist uncontrollable flaws in the manufacturing of a coin, and hence a head and a tail are no longer equally likely to happen. In this scenario, a number p can still be assigned to the event of a head as its probability, but this value p cannot be determined by the classical definition. Only the frequency approach can be applied in such a scenario to obtain an estimation/approximation of the value p by a large number of repeated trials and observations (Mood, 1974; Papoulis, 1991).

2.4.3 The law of large numbers

An implicit assumption underlying the frequency approach is the *law of large numbers* (LLN). LLN is an important theorem in probability theory and is one of the most important principles employed in statistical analysis, which can roughly be described as the law that the arithmetic mean of the results obtained from a large number of experiments almost surely converges to the expected value as the number of repeated trials approaches infinity (Mood, 1974; Grinstead and Snell, 1997).

In the context of relative frequency, LLN implies that as long as the sample of observations is sufficiently large, the relative frequency of a particular event is approximately equal to the probability of the event. Here, 'sufficiently large' means the sample is large enough so that extreme values cancel each other out. It should be noted that another underlying assumption of relative frequency is also important: the experiment from which the observations are obtained should be *repeatable*. That is, the experiment should be able to be repeatedly conducted under the same (or quite similar) conditions. The reason why this underlying assumption (of the frequency approach) should be carefully taken into account is further explained in Chapter 3.

2.4.4 Other interpretations of probability

Although widely applied in engineering and scientific research, the frequency approach is only a branch of probability theory: there are many other interpretations of probability that cannot be ignored. These include logical probability, subjective probability, and propensity (Hájek, 2012). Although each of these interpretations (including the classical and frequency interpretations) seems to be able to capture some crucial insight into the probability concept, none of them is flawless. Therefore, it may be more appropriate to treat these different interpretations as complementary.

Generally speaking, the major controversies between these different interpretations lie in whether probabilities 'live in the world' or 'live in the mind' and to what extent probabilities are objective/subjective (Parmigiani and Inoue, 2009). In terms of generality, classical probability is the narrowest due to the strict restrictions placed on symmetry and equipossibility, *subjectivism* is the widest, and *frequentism* lies in between placing moderate restrictions on repeatability and randomness (Figure 2.4).

It should be noted that the perspective of frequentism (i.e. the frequency interpretation) is adopted throughout this thesis.

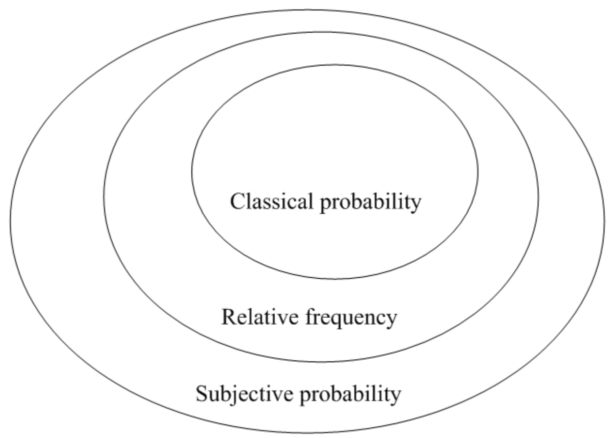


Figure 2.4 The generality of typical interpretations of probability

2.5 Fundamentals of railway timetabling and operations

Railway systems are highly complex systems that require the cooperation of various parties (e.g. passengers, train operators, infrastructure managers, public authority, and the general public) and involve a large number of interdependent production processes (e.g. line planning, timetabling, dispatching, track maintenance, passenger information, etc). This section is not aimed at providing a detailed and comprehensive introduction to all aspects of railway systems. Instead, it tries to provide an introduction to the key concepts in railway timetabling and operations that would help the understanding of the algorithms presented in subsequent chapters.

2.5.1 The hierarchy of railway planning processes

In order to guarantee smooth daily operations, a railway system needs to be carefully planned in advance. Railway planning involves a series of interrelated steps from demand estimation to real-time traffic control, and these steps can generally be categorised into three stages: strategic planning, tactical planning, and operational planning (see Figure 2.5). These steps are, however, not strictly separated and can influence each other. Note that this section is focused mainly on passenger transport.

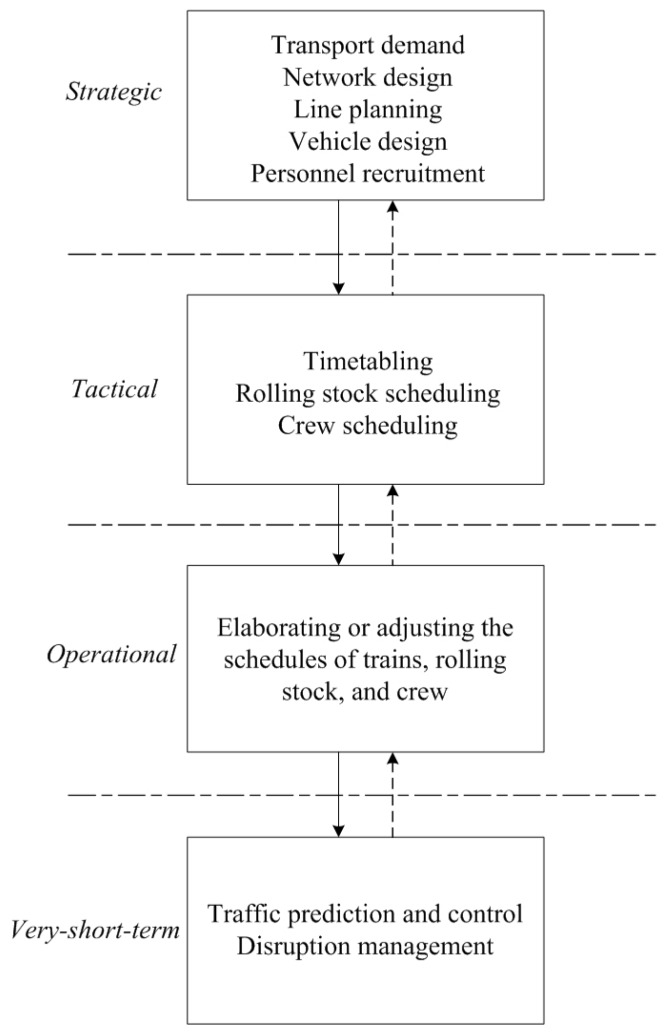


Figure 2.5 The hierarchy of railway planning [Source: Author. Based on the relevant literature including Goverde (2005), Huisman et al. (2005), Watson (2008), Andersson (2014), and D'Ariano et al. (2014)]

Strategic planning often happens well before trains are placed on tracks. Large and long-term investments are typically involved in this stage to construct new infrastructure, producing new rolling stock, hiring new staff, etc. (Goverde, 2005). And the major objective of this stage is to determine where and how tracks and lines should be built/designed, based on the estimation of market demand (Andersson, 2014).

Tactical planning is mainly concerned with the allocation of railway resources for the intermediate planning horizon. One of the major tasks in this stage is timetable construction: which trains should be allocated to which tracks during which time slots. This is not an easy task: timetable constructors need to simultaneously consider and balance the requests from different train operators and the requirements of track maintenance. Meanwhile, efficiently allocating rolling stock and scheduling crews are supposed to happen at this stage, both of which should also be carefully taken into account in timetable construction.

Operational planning and *Very-short-term planning* deal with short-term perturbations in a railway system. Since railway systems are highly complex, any mistake/malfunction in any operational process is likely to interrupt the smooth functioning of the whole system and lead to delays and disruptions (Yuan and Medeossi, 2014). Therefore, it is necessary for a railway system to have some mechanism to intervene in a disturbed situation and control/reduce the impact of the interruption. Existing mechanisms include local dispatching, network-level traffic control, shunting, and short-term rescheduling. While the major task of dispatching and traffic control is to resolve conflicting train paths during perturbations, shunting and rescheduling are mainly for the management of predictable variations in daily operations (e.g. peak/off-peak demand, engineering works, etc). For further details about each of these mechanisms, it is recommended to refer to Goverde (2005).

It should be noted that the planning stages and steps described above are not strictly in chronological order: they together form a feedback loop. For example, some of the problems in operational planning (e.g. the railway network is very sensitive to perturbations) would be likely to force timetable designers to consider improving the existing timetable, while other operational problems (e.g. there is a shortage of rolling stock or crews) would be likely to force decision makers to consider increasing investments in rail.

2.5.2 Timetabling terminology

A timetable is a rail operator's promise to its potential passengers about how train services are planned. The timetabling (timetable construction) process is often a complicated process that involves a trade-off between efficiency, safety, regularity, and conflicting interests.

A *master timetable* is a long-term timetable that is established for all traffic within a railway network during a given time period. European passenger railways usually adopt *periodic timetables*. Based on a periodic timetable, a train line is operated at regular intervals between different hours of a day and between different days. An advantage of periodicity is that it makes a timetable easy to memorise (see Figure 2.6).

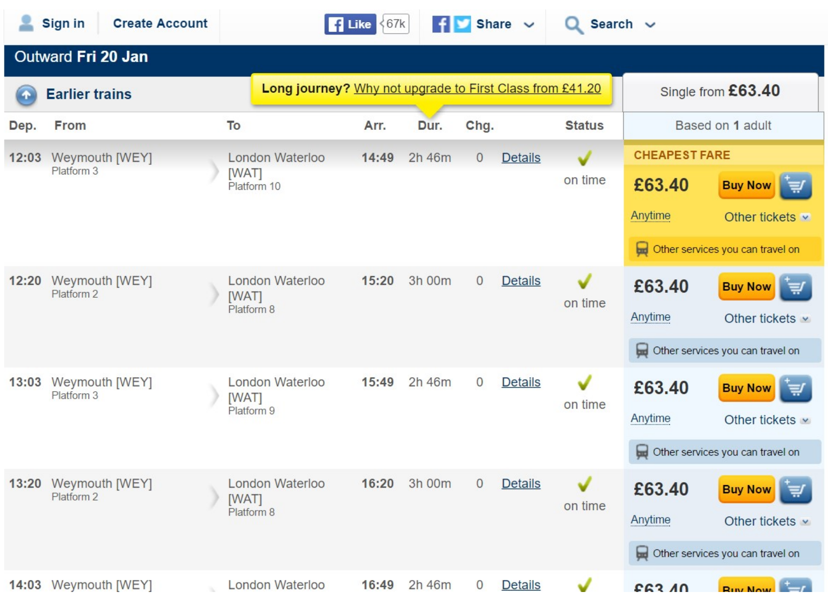


Figure 2.6 An example of periodic timetable (Source: www.nationalrail.co.uk, accessed 18 Jan 2017)

Due to periodicity, a long-term (i.e. yearly, or six-months in Britain) timetable for a train line can be constructed from a *basic hour pattern* (BHP), which can be visualised by a *train path diagram* (also called time-distance diagram) (see Figure 2.7). Note that a long-term timetable is often not strictly periodic and is not constructed by simply copying and concatenating BHPs: a BHP needs to be adapted to different time periods taking into account daily and hourly fluctuations in traffic demand (Goverde, 2005). Additional modifications to the long-term timetable might be needed due to maintenance and special events, which results in the *daily timetable* for each day during the timetable period.

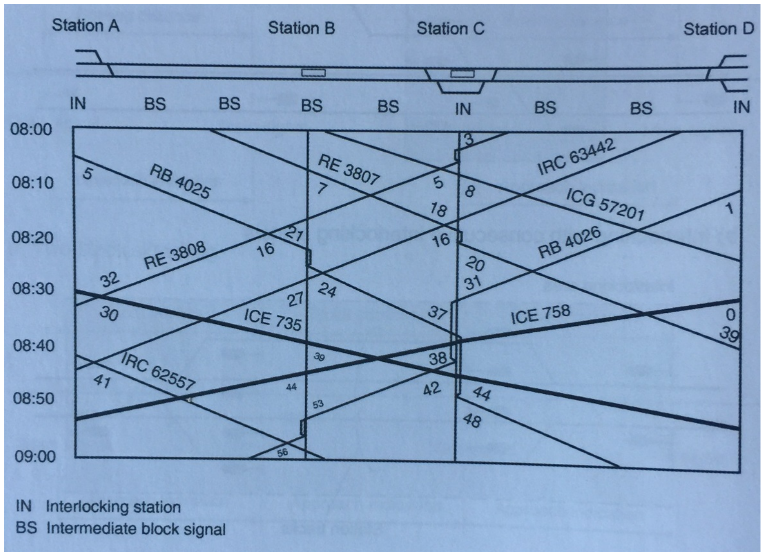


Figure 2.7 An example of train path diagram (Source: Pachl (2014))

1W58 1120 Weymouth to London Waterloo
South West Trains service departing on 17th January 2017

Schedule Information
- **WTT** schedule UID **W35039**, identity **1W58**
- Runs **SSuX** between **12/12/2016** to **25/08/2017**
- Service code **24620204**, headcode **9242**
- Express Passenger

Operational Information
- Schedule from **ITPS**
- Timed for **100mph** max
- **Electric** Multiple Unit

Passenger Information
- Retail Service ID **SW9242**
- Seating: first & standard
- Reservations available
- Trolley service from Bournemouth

Realtime Status
- TRUST ID **861W58MI17**
- Running as **1W58**
- Activated **17/01/2017 10:20**

Mileage				WTT		GBTT		Realtime			Route		Allowances		
M	Ch	Location	Pl	Arr	Dep	Arr	Dep	Arr	Dep	Dly	Line	Path	Eng	Pth	Prf
0	0	Weymouth [WEY]	2		1120		1120		1119½	RT					
0	28	Weymouth Jn		pass	1121			pass	1120¾	RT					
2	33	Upwey [UPW]	1	1124	1124½	1124	1124	1123¾	1124¾	RT					
6	49	Dorchester Jn		pass	1130			pass	1130	RT					
6	74	Dorchester South [DCH]	1	1131	1133	1131	1133	1130¾	1132½	RT					½
12	40	Moreton (Dorset) [MTN]	1	1139	1139½	1139	1139	1138½	N/R	RT					
16	75	Wool [WOO]	1	1144½	1145½	1145	1145	1145	1146	RT					
20	67	Worgret Jn		pass	1150			No report							
21	74	Wareham [WRM]	1	1151½	1153	1152	1153	1151	1152	1E					

Figure 2.8 An illustration of working timetable (Source: www.realtimetrains.co.uk, accessed 18 Jan 2017): the column under 'WTT' is the working timetable and 'GBTT' the published timetable.

Another pair of related concepts that need to be distinguished is published timetables and working timetables. In the context of British railways, a *published timetable* (i.e. the National Rail Timetable) is updated on a half-yearly basis, which contains information about all train services during a given period. It can be viewed as a promise from British rail industry to British rail passengers on the scheduled arrival and departure times, service frequency, planned journey times, availability of direct services, and transfer times and number of transfers when direct services are unavailable. By contrast, a *working timetable* contains more detailed information about planned train movements (e.g. train identifiers, freight train schedules, passing times at through stations, etc) than the corresponding published timetable, which is not for public use and is only circulated among rail industry professionals (Figure 2.8).

The planned running times (of a train) between consecutive scheduled stops in a published/working timetable are based on the corresponding *nominal running times*, which are the physically possible minimum running times. Normally, planned/scheduled running times are set slightly larger than the corresponding nominal running times, and the positive difference between a nominal running time and its corresponding scheduled running time is called *running time supplement* (also called running time margin or allowance) (Vromans, 2005). According to UIC (2000), running time supplements can be incorporated into a timetable in the following three ways: 1) distance dependent supplements [mins/km], 2) running time dependent supplements [%], and 3) fixed-size supplement per station/junction [mins]. Dutch railways adopt an industry standard of 7% for all passenger services (Goverde, 2005; Vromans, 2005). That is, 7% of the nominal running times are added into timetables as running time supplements. German railways utilise running time supplements ranging from 3% to 7%, depending on types of trains and track characteristics (Goverde, 2005). In Britain, running time supplements are not explicitly defined but are included in the timetables (Vromans, 2005). According to Goverde (2005), the addition of running time supplements can serve the following three purposes: 1) allow a slower speed profile under less favourable conditions such as bad weather, electrical current fluctuations, drivers behaviour, etc; 2) serve as recovery time to reduce the impact of departure delays; and 3) enable more energy-efficient running by coasting.

Whereas running time supplements are utilised to enable a train to make up small delays, *buffer times* are added into a timetable to prevent delay transmission between different trains (Pachl, 2014). Two major types of buffer times can be distinguished: 1) *headway buffer times* (i.e. the scheduled headway between two trains should include extra time to compensate for small delays); 2) *transfer buffer times* (the scheduled transfer time needs to include extra time to prevent the transmission of delays during the transfer process of passengers and/or crews).

It should be noted that although the exploitation of running time supplements and buffer times could to some extent improve the reliability and robustness of a railway system, the abuse of them could also lead to increased track and station consumption and hence result in unnecessary capacity losses. Hence, there often exists a balance between efficiency and robustness in the practical timetabling process (Vromans, 2005).

2.5.3 Rail data collection

Compared with other means of transport, rail transport has its unique characteristics: heavy vehicles run on fixed tracks at a considerably high speed, and long braking distances result from low friction between rails and wheels (Goverde, 2005). Therefore, highly reliable *safety systems* should be installed to prevent derailments, collisions between vehicles, and casualties. The safety subsystem of a railway system is embodied by the comprehensive *signalling (sub)system*, which is mainly composed of train detection devices, trackside signalling (automatic/controlled), and cab signalling (automatic). For more details about the safety and signalling systems in European railways, it is recommended to refer to Bailey (1995), Goverde (2005), and Pachl (2014).

Signalling devices are not only employed to prevent accidents but also used to record and monitor train movements in real time, which facilitates rail data collection and the statistical analysis of a railway system. In modern railways (in most European countries), there are generally two sources of train movements data: Train Describer and Train Event Recorder. The *train describer* system holds a database receiving and containing plentiful information about the real-time position of trains at the signal level, which functions as an important tool for traffic management, track supervision, automatic route setting, and passenger information (Bailey, 1995). In Britain, the train describer data are managed by the infrastructure manager (i.e. Network Rail), and ORDW (2016) provides more details about the format of these data. *Train event recorders* are widely used in European railways, which are analogous to flight recorders (commonly known as black boxes). Integrated with other car-borne systems, they enable enhanced diagnoses and controls (e.g. automatic warning, emergency braking, etc). Although train event recorder data are mainly used for accident analysis and prevention, they are also used to monitor train performance (Yuan and Medeossi, 2014).

2.5.4 Performance measures

The performance of rail transport can be evaluated from different standpoints of different parties (e.g. customer-oriented, operator-oriented, government-oriented, etc) and from a wide range of different dimensions (e.g. economy, efficiency, reliability, safety, environment-friendliness, etc). Clearly, there are no standardised performance measures that apply to all parties and situations. But it is widely accepted that punctual, reliable, and fast transport of people and goods at minimum cost would help increase the competitiveness of rail transport (Martin, 2014). Here (and in subsequent chapters of this thesis), the focus is mainly on reliability and punctuality. More specifically, this subsection is mainly aimed at introducing the (reliability and) punctuality measures that are currently in use in most European railways.

Reliability and punctuality are a major concern of both rail passengers and rail operators (Yuan, 2006; ATOC, 2013). While reliability has a much broader meaning, punctuality is generally used to describe how late an average train arrives (Rietveld et al., 2001; Olsson and Haugland, 2004; Vromans, 2005; Preston et al., 2009). For the convenience of quantitative analysis, European railways often adopt heuristic measurements such as presenting punctuality as the percentage of trains that run within a predefined level of acceptable deviation (e.g. 5 mins) from the official timetable (Olsson and Haugland, 2004; Preston et al., 2009). These heuristic measurements, however, tend to omit a lot of realistic issues and hence are only rough estimations at the macroscopic level. A lot of information is hidden about punctuality at intermediate stops (Olsson and Haugland, 2004; Martin, 2014) due to the statistical method employed by rail operators (i.e. punctuality is often measured only at terminating or large major stations). And since the performance indicators currently in use by operators are mostly train-oriented (supply-oriented), they tend to overestimate the service quality experienced by travellers (Harris, 1992; Rietveld, 2005; Weston et al., 2006; Carrasco, 2012; Harris et al., 2013). Although these heuristic measurements are in themselves problematic, they could to some degree help rail operators monitor the overall performance of their train services and help public authority formulate performance-related policies: underperforming rail operators would be confronted with fines, and rail operators are responsible for direct compensation to rail passengers when significant delays/disruptions happen (Rietveld, 2005; Preston et al., 2009).

2.6 Related concepts in Decision Theory

Decision theory is mainly concerned with decisions. More concretely, it is a theory about goal-directed behaviour in the presence of choices/options (Hansson, 1994). Researchers from many disciplines (e.g. economics, statistics, sociology, psychology, etc) have contributed to the development of decision theory. While the domain of decision theory includes a wide range of relevant topics, this section is only aimed at providing a brief introduction to several relevant concepts to this thesis. For a more rigorous and comprehensive introduction to decision theory, it is recommended to refer to e.g. Parmigiani and Inoue (2009) and Bradley (2014).

2.6.1 The classification of decision-making

Decision theory is built upon several basic concepts: alternatives, states of nature, and outcomes. A decision maker is assumed to be confronted with a finite set of mutually exclusive *alternatives*, each of which is a course of action that can be taken by the decision maker at the time of decision making. While a decision maker might have, in the process of decision making, some background information about some of the various extraneous factors that are beyond his/her control, there often exist a number of unknown extraneous factors. These unknown extraneous factors can be summarised into a number of scenarios, called *states of nature* in the terminology of decision theory. With these two concepts (i.e. alternatives and states of nature) at hand, the *outcome* of a decision can be defined as the combined effect of the chosen alternative and the realised state of nature.

certainty	deterministic knowledge
risk	complete probabilistic knowledge
uncertainty	partial probabilistic knowledge
ignorance	no probabilistic knowledge

Figure 2.9 The categories of decision problems (Source: Hansson (1994))

Dependent upon how much information is available about the states of nature in decision making, the various decision problems can generally be categorised into four groups: *decision making under certainty*, *decision making under risk*, *decision making under* *uncertainty*, and *decision making under ignorance* (the amount of available information is decreasing from left to right) (see Figure 2.9).

2.6.2 Expected utility

In order to make decisions (i.e. choose between a set of alternatives), it would be helpful to have some value standard (measurement) at hand for determining/evaluating how good the outcome of a particular alternative is and then compare alternatives based on this standard. A commonly adopted value standard is called *utility*, which can be defined as units of human happiness in the terminology of moral philosophy. Many economic or utilitarian moral theories are based on the rule of *utility maximisation*, meaning that a decision maker chooses (one of) the alternative(s) that maximises his/her utility.

Expected utility (EU) is the mainstream approach to decision making under risk (refer to the classification of decision problems in the previous section), which assumes that the probabilities of all states of nature are known. According to expected utility theory, each alternative can be assigned a value representing the weighted average of the utility values under different states, and the weights adopted are just the probabilities of these different states. The rule of maximisation in expected utility theory is called *maximum expected utility* (MEU), which means that a decision maker chooses (one of) the alternative(s) that maximises his/her expected utility.

2.6.3 Principle of indifference

In reality, complete probabilistic knowledge is often unavailable about states of nature and decision makers would have to make decisions under uncertainty or under ignorance. In these situations, the *principle of indifference* (also called *the principle of insufficient reason*) is often employed to simplify a decision problem and reduce ignorance/uncertainty to risk.

The principle of indifference (POI) is as old as probability theory that is introduced previously in Section 2.4. In fact, the classical definition of probability (in subsection 2.4.1) can be viewed as based on POI: for a finite set of N mutually exclusive outcomes, if there is no reason to believe that one outcome is more likely than another to occur, then the N outcomes should be treated as equipossible, each of which has a probability of 1/N.

The limitations of POI lie mainly in two aspects. Firstly, the result obtained from the application of POI depends on the partitioning of the alternatives and hence whether the structure of the states of nature is symmetrical should be checked before applying POI. Secondly, POI is not applicable to decision making under *absolute* ignorance: neither is the probabilistic knowledge about states of nature available, nor is the knowledge about the states of nature themselves is available (i.e. whether a particular state exists is unknown).

Although POI is not a perfect solution to decision problems under ignorance, it is widely utilised in scientific research and engineering applications.

2.7 Summary

This chapter introduces a considerable number of relevant concepts to this thesis that come from several different but interrelated academic disciplines. The fields involved include Algorithm Engineering, Probability and Statistics, Railway Timetabling and Operations, and Decision Theory. Although some of these concepts are not directly touched upon in this thesis, a good understanding of them would be helpful to understanding the subsequent chapters. Some of the briefly introduced concepts are to be further explained or illustrated in subsequent chapters when they are applied to specific scenarios. Moreover, some of the concepts in the relevant fields are omitted in this chapter to avoid confusion. But they are to be introduced in subsequent chapters with the aid of specific contexts.

Chapter 3

Critical Routes: a weak point of existing journey planning systems

3.1 Introduction

This chapter is centred on the introduction to the concept of critical routes, which can roughly be described as those transfer-involved, long-distance, and delay-sensitive routes within a given railway network. The subsequent sections are organised as follows. Section 3.2 gives a general introduction to the status quo of passenger information in British railways. Following that, Section 3.3 describes the problem that currently exists in the pre-trip information about those transfer-involved routes. This section is followed by a detailed algorithmic-level explanation of why it is difficult to effectively deal with those transfer-involved routes using existing journey planning technologies in Section 3.4. In order to efficiently identify those problematic transfer-involved routes within a railway information system, a screening algorithm is developed and presented in Sections 3.5 and 3.6: definitions of several introduced concepts are presented in Section 3.5, and the algorithm together with its explanations can be found in Section 3.6. After that, the applicability of the developed screening algorithm is illustrated in Section 3.7 by a case study of the National Rail timetable currently in use in Britain's passenger rail system. Section 3.8 conducts a further investigation into the train delay data briefly described in Section 3.7 to gain additional knowledge about passenger train delays in British railways: the obtained statistical models and their interpretations are presented in this section. Section 3.9 concludes this chapter.

3.2 The status quo of passenger information in British railways

The past decade has seen a boom in the Internet's popularity. Statistics have shown that three billion people around the world (3/7 of the population) are now connected to the Internet

(Meeker, 2016) and in Great Britain 23.7 million households (89% of the total) have access to the Internet (ONS, 2016). The new wave of Mobile Internet (i.e. fast and stable connections to the Internet via smart phones, tablets, and other mobile devices) further defines what we can expect from this 'digital age'(Lyons, 2015; Meeker, 2016; ONS, 2016).

Figure 3.1 Snapshots of some of the TOCs' websites

Figure 3.2 TOCs in social media

The impact of Internet-related technologies on traditional industries is remarkable, and the rail industry is no exception. In Great Britain, apart from the National Rail Enquiries (NRE) website (see Figure 1.3 in Chapter 1), most Train Operating Companies (TOCs) have their own versions providing online information and ticketing services (Figure 3.1). Besides, accounts or homepages of rail companies can also be easily found on popular social media such as Facebook, Twitter, etc. for marketing and information purposes (Figure 3.2).

Although how much impact these information and communications technologies (ICTs) can have on rail demand and patronage remains an open question, it is generally believed that providing passengers with timely and reliable travel information plays an important role in improving customer experience and stimulate rail use (Chorus et al., 2007; Lyons et al., 2008; ATOC, 2013; Ben-Elia and Avineri, 2015; RRUKA, 2015).

The efforts Great Britain's rail industry has made on passenger information can partly be seen from a wealth of open data on train operations available from the Internet (more details can be found on Open Rail Data Wiki[2]), which enables the public to participate in improving rail travel information. Several travel information websites (e.g. Open Train Times[3] and Realtime Trains[4]) and a number of mobile applications are built on these open data, either directly or indirectly.

Basically, the various forms of passenger information can be classified into two broad categories: static pre-trip information and dynamic real-time information. The former includes printed train timetables and timetable-based journey planning web applications such as National Rail Enquiries (NRE) (see Figure 1.3). The latter ranges from in-station displays and broadcasts to the diverse officially and unofficially deployed mobile applications such as National Rail Travel App, Realtime Trains, etc. In reality, however, the quality of passenger information is not always guaranteed, especially in the domain of pre-trip information. And there seems to be a lack of a bridge between static pre-trip information and dynamic real-time information due to the asynchrony between these two relatively independent domains.

3.3 The problem of pre-trip information about transfer-involved routes

Direct rail routes (lines) are often characterised by higher transport demand and more frequent train services, which naturally receive more attention from rail operators. After all, if

[2] http://nrodwiki.rockshore.net/index.php/Main_Page
[3] http://www.opentraintimes.com/
[4] http://www.realtimetrains.co.uk/

these direct routes are poorly performed (in terms of punctuality and reliability) and poorly serviced (e.g. poor information services), the corresponding rail operators would have to be confronted with fines by the public authority and compensation to the passengers.

Transfer-involved rail routes[5] receive, however, much less attention from rail operators due to various reasons. In Great Britain, the organisational structure of the rail sector is characterised by the coexistence of a number of relatively independent Train Operating Companies (TOCs): if a route involves several different train lines managed by different TOCs, then it is difficult to determine who should take the responsibility for undesirable performance and services. On the other hand, even if the coordination of different rail operators is not a problem, it is still technically challenging to deal with these transfer-involved routes due to the limitations of existing planning and information technologies. Since the topic of this chapter (and this thesis) is limited to passenger information, only the information-related problem of transfer-involved rail routes is described in this section.

The gap between static pre-trip information and dynamic real-time information has been mentioned in the previous section (Section 3.2). Although this gap is negligible in many cases, it becomes non-negligible when a given (recommended) itinerary involves delay-sensitive transfers (interchanges). Figure 3.3 provides an illustrative example of such non-negligible problems in current passenger information systems.

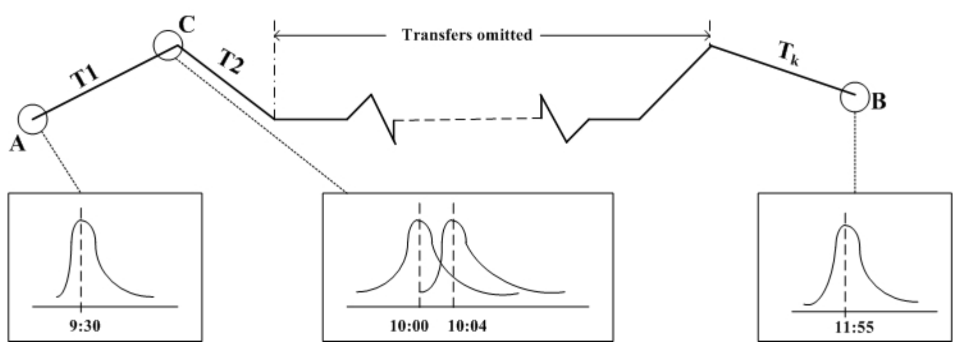


Figure 3.3 An illustration of a rail journey involving a number of transfers

Suppose one day a rail passenger wants to go from Station A to Station B, and he/she provides the pair of A and B as well as an expected departure time to an online journey planning system. Then the system returns a recommended itinerary as follows (c.f. Figure

[5] the exact meaning of which is to be clarified later in this section.

3.3): take Train 1 (denoted as T1) departing from A at 9:30 (a.m.) to arrive at C at 10:00, then transfer at C from T1 to T2 (Train 2) departing from C at 10:04 to arrive at…(instructions for intermediate transfers are omitted)…and finally take Train k (T_k) to arrive at B at 11:55 (a.m.).

From this illustrative example, we can catch a glimpse of some key characteristics of pre-trip timetable information about rail journeys that involve transfers: when provided a pair of origin and target stations and an expected departure time, a computer-based journey planning system looks up all the relevant train timetables according to some journey planning algorithm (refer to Section 2.3) to generate an 'optimal' journey plan in terms of total journey time (TJT, not including the access/egress parts) or a set of *Pareto-optimal* (i.e. non-dominated) journey plans in terms of TJT, number of transfers (NoT), fare, etc. In practice, however, the existence of train delays and cancellations tends to make the pre-trip timetable information about the arrival and departure times along the recommended route(s) *unreliable* and the arrival at the target station *unpunctual*. Rather than deterministic single values (e.g. 9:30, 10:00, etc. in the example), these arrival/departure times may look more like stochastic distributions (see Figure 3.3).

Here, differentiation is made between *nominal arrival/departure time* and *actual arrival/departure time*. Nominal arrival/departure time (NAT/NDT) refers to some planned arrival(departure) time in a long-term timetable that is adopted by a journey planning system to process well before a given journey begins. Actual arrival/departure time (AcAT/AcDT) means some recorded arrival/departure time for a given train at a given station after the train service has finished. The TJT calculated from NATs and NDTs is called *nominal journey time* (NJT, e.g. 2h25m in the above example), and the TJT calculated from AcATs and AcDTs is called *actual journey time* (AcJT).

Based on the following three observations: (1) there is often a significant discrepancy between the NJT and AcJT of a transfer-involved rail journey, (2) in reality rail journeys involving transfers are more prone to train delays and cancellations than those involving no transfers, and (3) some of those transfer-involved rail journeys are much more sensitive to train delays and cancellations than the others, the following questions arise: How many journeys particularly prone to delays and cancellations exist in a given railway network? How to efficiently identify them? If such journeys do exist, then how to exploit the available information tools to reduce the negative effects of unreliability and unpunctuality?

The answers to these questions are not that straightforward, and the solution proposed to the problem of pre-trip information about transfer-involved rail routes is to be detailed in subsequent sections and chapters.

Before leaving for the next section, a clarification is also needed to make about a set of closely related concepts – journey, itinerary, transfer, leg, and route – to avoid misunderstandings in subsequent sections and chapters.

In the context of this thesis, the concepts of a *journey* and an *itinerary* are largely interchangeable, both of which are defined on the dimensions of both time and space and correspond to a specific sequence of involved trains (legs) and the corresponding sequence of transfer stations. Moreover, the word *journey* (or *itinerary*) is in most cases linked with an unrealised (not-yet-achieved) plan in this thesis and hence is equivalent to the phrase of *journey plan* (or *itinerary plan*). Note that two variants of the concept of journey/itinerary (i.e. *itinerary template* and *reconstructed itinerary*) are to be introduced in Section 4.3. Further explanation and clarification is to be made in Section 4.3 with the aid of specific contexts.

Another pair of interrelated concepts is *transfer* and *leg*. A *transfer* is between two consecutive legs of a journey/itinerary, while a *leg* corresponds to a specific train connecting two transfer stations. In this description/definition, a 'transfer station' can be either intra-modal or inter-modal. The concrete example in Figure 3.3 may help understand this: the k involved trains in the figure can be viewed as legs, linking the (k-1) intra-modal transfer (railway) stations and the two inter-modal transfer stations (i.e. the origin station A and the destination station B). It can be seen from the above descriptions that the concepts of *transfer* and *leg* are also time- and space-specific, for they are essentially the components of a journey/itinerary. However, it should be noted that these two notions can have more generalised meanings in subsequent sections and chapters: in certain contexts (in the remaining of this thesis), the word *transfer* is employed to represent a *generic transfer* and the word *leg* is utilised to represent a *generic leg*. A *generic* transfer/leg is defined only on the dimension of space (more precisely, it is also partially defined on the dimension of time: see the clarification of the term *route* in the next paragraph), referring to a set of specific transfers/legs that follow the same pattern (in terms of space) but occur at different hours of day and different days of week. Further illustration and clarification can be found later in subsequent sections and chapters, with the aid of specific contexts.

The notion of a *route* need also be clarified. Often, the term *route* is related to road networks and is defined only on the dimension of space. As previously mentioned in Subsection 2.3.2, a spatial description/definition of a route is often enough in the context of (private) road transport due to the fact that car owners have considerable freedom to choose a desired departure time (and also a desired arrival time), without the constraint of vehicle service providers and infrastructure managers. In contrast, a *rail route* is constrained by planned timetables detailing the opening and closing times of the relevant train services and tracks, which partially incorporates an additional dimension of time. Here, 'partially' is used to emphasise that although the temporal dimension is introduced a rail route is usually referred to in a generic way: it can be viewed as an abstraction of a set of relevant train services connecting two given railway stations. Often, the notion of a rail route is related to a *direct route*, the source and target stations of which are connectable by a single *line* (a railway line corresponds to a set of trains that follow a specific (periodic) timetable). In the context of this thesis, a novel notion of a *transfer-involved rail route* is introduced, which can be viewed as an extension of the notion of a *rail route*: if a given target station is reachable from a given source station but no direct route exists between them, then the chronologically ordered set of the relevant legs and transfers is called a *transfer-involved rail route* between the two stations.

3.4 Existing journey planning algorithms: intelligent or not?

3.4.1 The 'art' of criteria and parameters selection

As has been described in Section 2.3, the state-of-the-art journey planning algorithms are all highly-developed and well-designed and are much more able than those early versions of shortest-paths algorithms to model and deal with realistic journey planning. Despite the significant improvements in terms of effectiveness and efficiency (compared with previous generations of routing algorithms), the current journey planning algorithms are, after all, built on mathematical models with predefined rules, criteria, and parameters. And due to the complexity in transfer-involved journeying and the quite different preferences of different rail passengers, the quality of the computed results of a journey planning system is heavily dependent upon the criteria and parameters adopted by the system. Here, in this subsection, an illustrative example is presented to show the subtleties in the choice of criteria and parameters. Note that although the illustrations employed in this section are mainly based on National Rail Enquiries (NRE), the phenomena revealed are common in the existing pre-trip journey planning systems. Since NRE has been among the most advanced around the world (c.f. Table A1 in Appendix A), the relevant technologies underlying NRE can be viewed as a reflection of the state-of-the-art journey planning systems in operation.

The example is a query with London Waterloo being the origin station, Exeter St David's being the destination station, and the desired departure time being 10:00 a.m. on Mon 23 Jan 2017. Two versions of the recommended itinerary list can be obtained, which are shown below: Figure 3.4 presents the version of NRE, while Figure 3.5 shows the version computed from a self-developed journey planning simulator by the author.

A lot of differences can be seen from these two pieces of information. Comparing between the two figures, the first impression may be that the simulator lacks the information about the fare and real-time status of services. But this is not the key point (because integrating fare information and real-time alerts into the simulator is theoretically not a difficult task, as long as detailed fare data and train status feeds are publicly accessible). A more significant difference lies in that the result set of the simulator is much larger than that of NRE, and the direct train service from London Waterloo to Exeter St David's with a departure time of 10:20 is omitted in the simulator (since the criterion of earliest arrival (c.f. Subsection 2.3.5) is given a higher priority than number of transfers in the simulator, the 10:20 direct service is excluded from the recommended list). This huge difference is, however, not that striking from a developer's perspective: the results for those routes involving transfer activities between train lines with different service frequencies are unavoidably sensitive to the predefined rules, criteria, and parameters due to the limitations of existing journey planning algorithms.

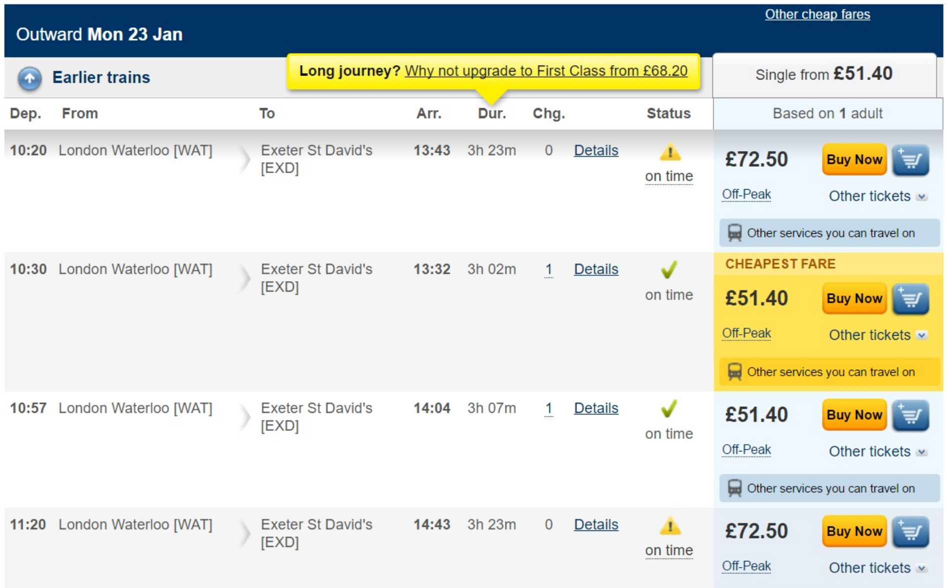
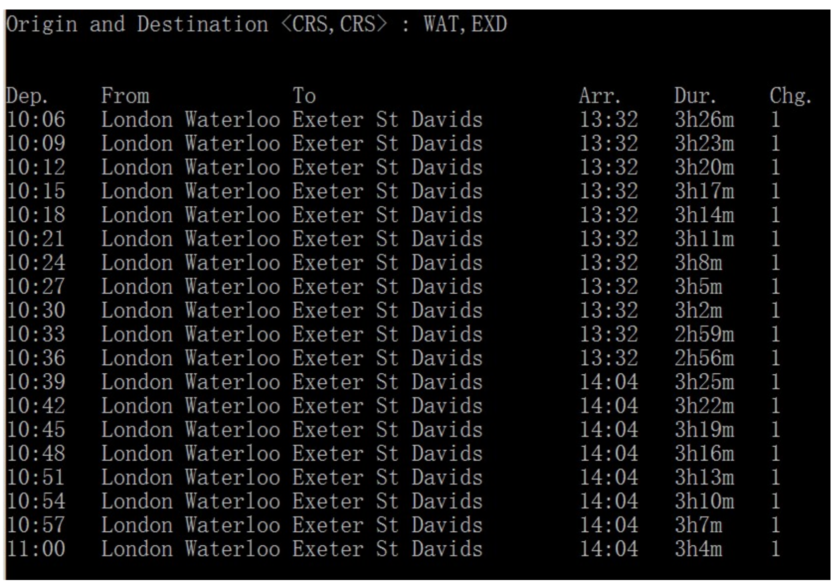


Figure 3.4 The version from NRE (accessed 20 Jan 2017)



```
Origin and Destination <CRS,CRS> : WAT,EXD

Dep.   From           To                      Arr.   Dur.    Chg.
10:06  London Waterloo Exeter St Davids       13:32  3h26m   1
10:09  London Waterloo Exeter St Davids       13:32  3h23m   1
10:12  London Waterloo Exeter St Davids       13:32  3h20m   1
10:15  London Waterloo Exeter St Davids       13:32  3h17m   1
10:18  London Waterloo Exeter St Davids       13:32  3h14m   1
10:21  London Waterloo Exeter St Davids       13:32  3h11m   1
10:24  London Waterloo Exeter St Davids       13:32  3h8m    1
10:27  London Waterloo Exeter St Davids       13:32  3h5m    1
10:30  London Waterloo Exeter St Davids       13:32  3h2m    1
10:33  London Waterloo Exeter St Davids       13:32  2h59m   1
10:36  London Waterloo Exeter St Davids       13:32  2h56m   1
10:39  London Waterloo Exeter St Davids       14:04  3h25m   1
10:42  London Waterloo Exeter St Davids       14:04  3h22m   1
10:45  London Waterloo Exeter St Davids       14:04  3h19m   1
10:48  London Waterloo Exeter St Davids       14:04  3h16m   1
10:51  London Waterloo Exeter St Davids       14:04  3h13m   1
10:54  London Waterloo Exeter St Davids       14:04  3h10m   1
10:57  London Waterloo Exeter St Davids       14:04  3h7m    1
11:00  London Waterloo Exeter St Davids       14:04  3h4m    1
```

Figure 3.5 The version from a self-developed simulator

More information about those transfer-involved itineraries could be found through a closer examination of the two involved legs: the first is a tube (metro) leg from London Waterloo to London Paddington, and the second is a direct rail line originating from London Paddington (Figure 3.6). Clearly, the train services connecting Paddington and Exeter are faster and more frequent than those directly connecting Waterloo and Exeter. And since it is convenient to go from Waterloo to Paddington due to the high-frequency tube services, this transfer-involved route may be favoured by some of the passengers. Therefore, rather than simply judge which of the two versions is better, it may be more appropriate to explain the discrepancy between the two result sets as the difference in travellers preferences (a developer is also very likely to be a rail passenger): if higher priority is assigned to direct services and conciseness of the result set, then the NRE version is the better representation; in contrast, if earliest-arrival and availability of options are the major concerns, then the simulator-generated version is better. Some may argue that the existence of these different versions can be resolved by developing a fully-customisable journey planning algorithm. In reality, however, there is always a trade-off between customisability and the complexity of the underlying model. Finding out a solution to this dilemma is very challenging based on current technologies.

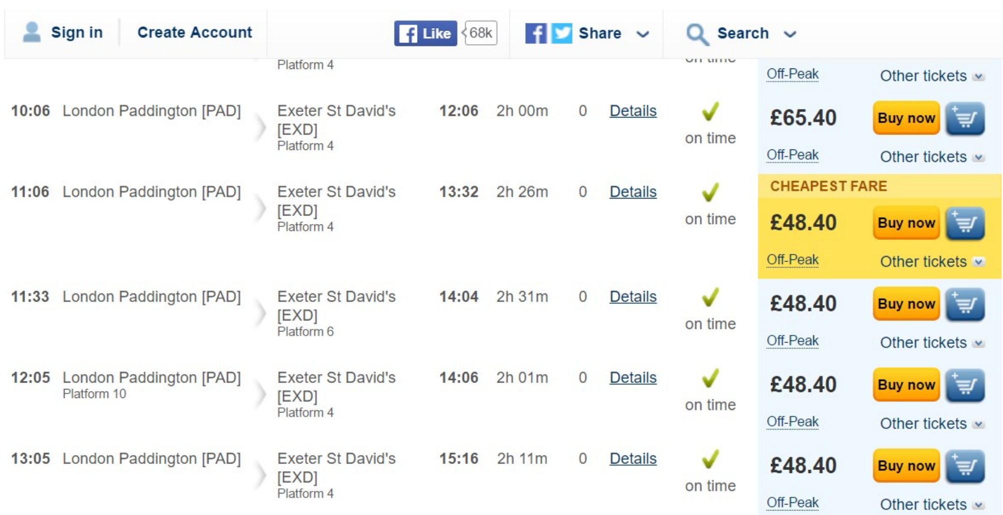


Figure 3.6 The timetable of the connecting leg (Source: NRE, accessed 23 Jan 2017)

3.4.2 The algorithmic-level mechanism of itinerary construction

Continuing the comparison between Figure 3.4 and Figure 3.5, another noticeable difference can be seen: while there are multiple itineraries having the same scheduled arrival time (13:32 or 14:04) in the simulator-generated version, the projection/relation between recommended itineraries and scheduled arrival times is one-to-one in the NRE version (only one itinerary corresponds to a scheduled arrival time 13:32, and the same for the scheduled arrival time 14:04). Moreover, the two itineraries with scheduled arrival times being 13:32 and 14:04 in the NRE version seem to have relatively late (compared with the available options in the simulator version) scheduled departure times (10:30 and 10:57, respectively).

This phenomenon is not just a special case, but is in the generality. Figures 3.7 – 3.10 gives another example.

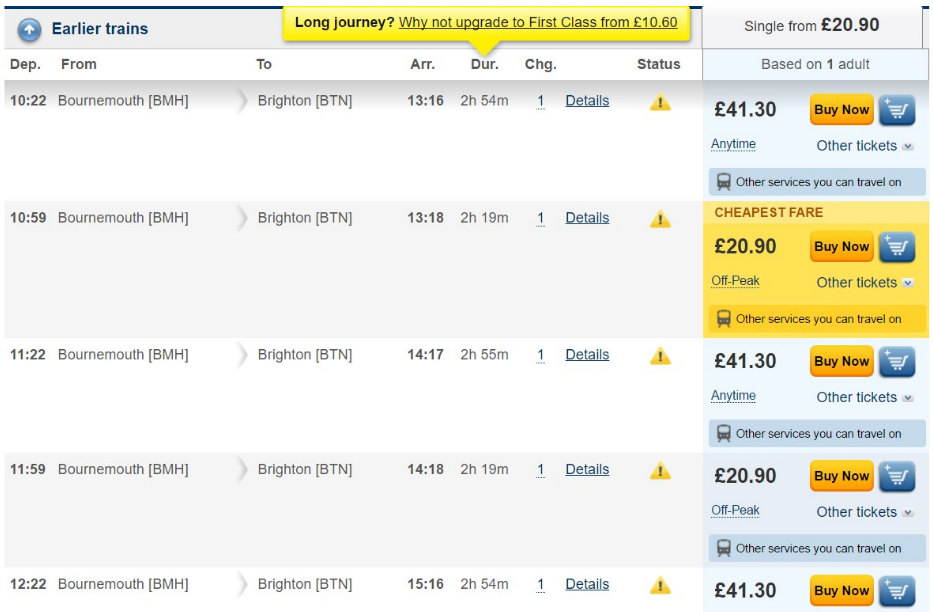


Figure 3.7 The recommended itinerary list for a journey from Bournemouth to Brighton (Source: NRE, accessed 23 Jan 2017)

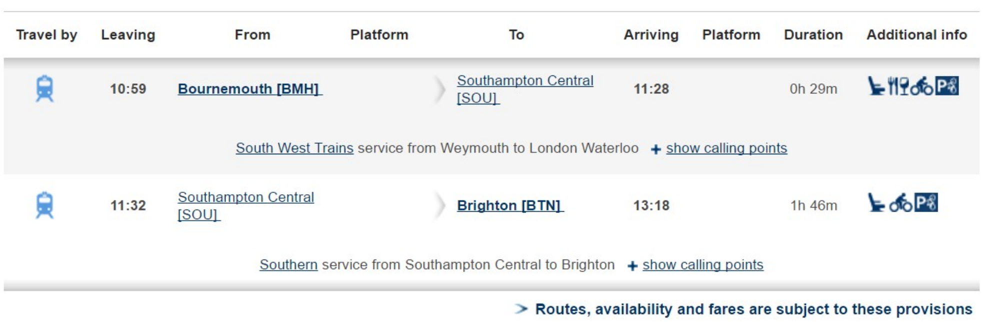


Figure 3.8 The adopted transfer plan by NRE for a route via Southampton Central (Source: NRE, accessed 23 Jan 2017)

This is a query about recommended journey plans from Bournemouth to Brighton (on 23 Jan 2017). There are generally two alternative routes for such a journey: via Southampton Central or via Clapham Junction. Due to the periodicity of the timetable, these itineraries can be grouped into two patterns (see Figure 3.7): those with a departure time of XX:22, a longer journey time, and a more expensive fare are via Clapham Junction, and those with a departure time of XX:59, a shorter journey time, and a cheaper fare are via Southampton Central.

Take the recommended itinerary departing at 10:59 for example (see Figure 3.8). Such a journey could generally be subdivided into three stages: the first stage is a ride from Bournemouth to Southampton Central, the second is an interchange (transfer) activity at Southampton Central, and the third is a ride from Southampton Central to Brighton. The transfer plan adopted by NRE for this itinerary (i.e. take the 10:59 South West Trains service for the first stage and take the 11:32 Southern service for the third stage) seems no problem, but a question arises if we take a closer look at each part (leg) of this route (see Figures 3.9 and 3.10): Why it is the 10:59 South West Trains service that is chosen as the feeder train? Why the other available options with more reserve for the transfer are not adopted (or at least displayed as alternatives)?

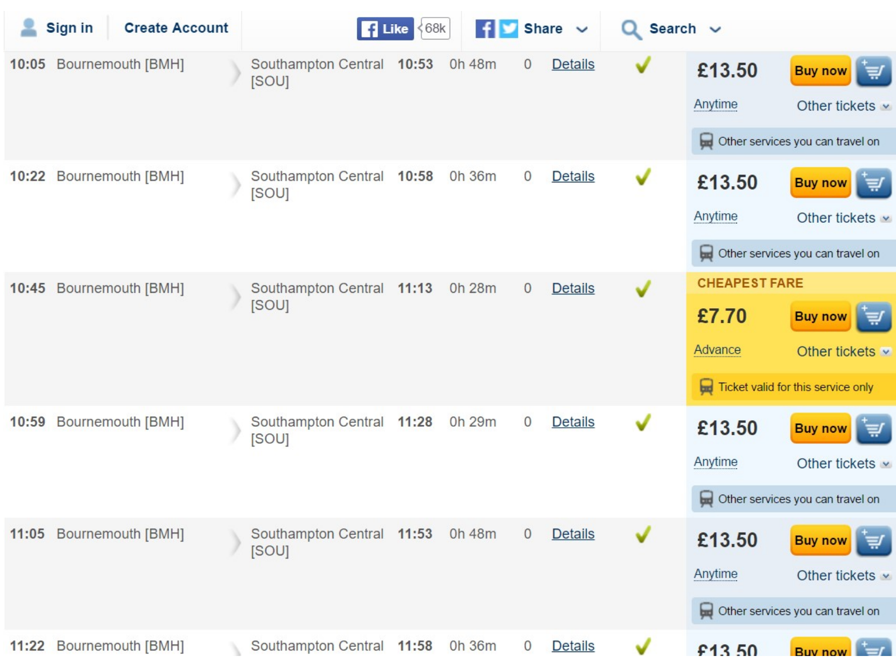


Figure 3.9 The available services for the feeder leg between Bournemouth and Southampton Central (Source: NRE, accessed 23 Jan 2017)

Figure 3.9 tells us that there are approximately four available options (corresponding to four different train services) per hour going from Bournemouth to Southampton Central. For example, four train services respectively departing at 10:05, 10:22, 10:45, and 10:59 are available between 10:00 and 11:00, and all of them are planned to arrive at Southampton Central before the scheduled departure time of the 11:32 Southern service that connects Southampton Central to Brighton (Figure 3.10). Since the connecting leg has less-frequent services (operated on an hourly basis), these four services (i.e. 10:05, 10:22, 10:45, and 10:59) can all be viewed as feeder trains to the 11:32 Southern train. However, only the 10:59 South West Trains train is chosen and displayed in the recommended itinerary list (see Figure 3.7). Apparently, NRE adopts an additional ‘latest departure’ (the 10:59 service is the latest among the four available feeder options) rule to earliest arrival (refer to Subsection 2.3.5) to achieve the conciseness of the recommended itinerary list.

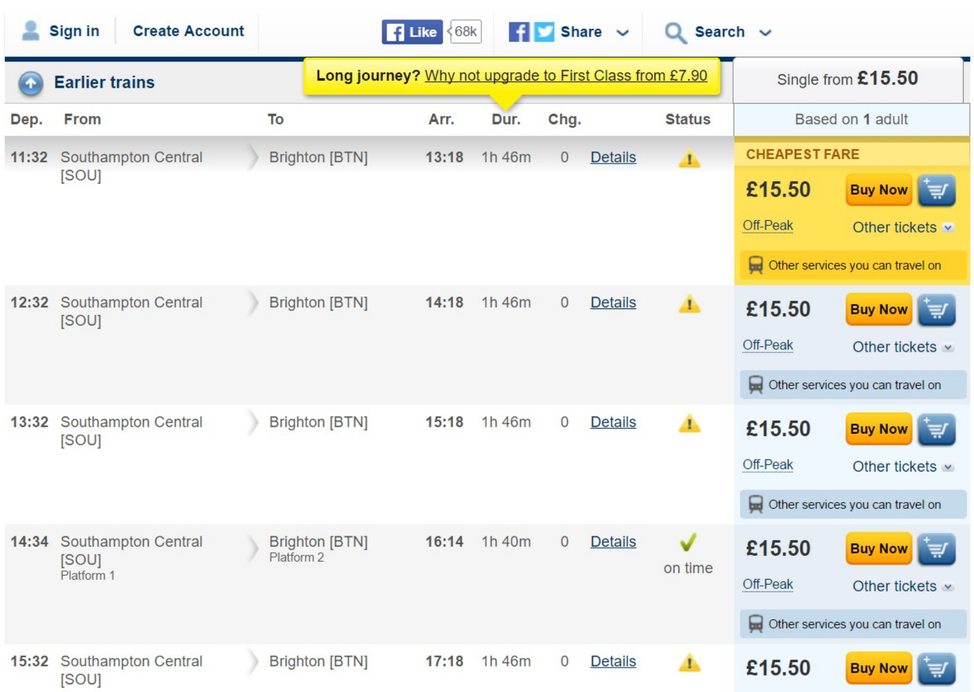
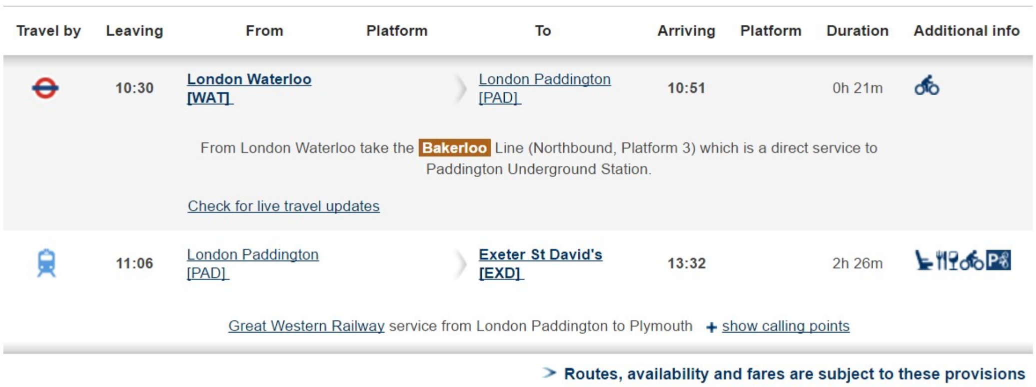


Figure 3.10 The available services for the connecting leg between Southampton Central and Brighton (Source: NRE, accessed 23 Jan 2017)

Looking back at the Waterloo – Exeter example in the previous subsection, we can see that the same rule (i.e. latest departure) applies. That is, in the NRE version (Figure 3.4) the two itineraries transferring at Paddington have the latest scheduled departure time(s) among the available feeder options. Some may argue that these two (i.e. departing at 10:30 and 10:57, respectively) do not follow the latest departure rule: according to the result set in the simulator version (Figure 3.5), the latest departure ones should be 10:36 and 11:00, respectively. That is, if the latest departure rule is applied to the result set in Figure 3.5, then the two itineraries remaining in the list should be the one with the scheduled departure time of 10:36 and the one with the scheduled departure time of 11:00, which are not the two itineraries adopted in the NRE version. This difference is, however, caused by the difference in the choice of parameters. NRE assigns a travel time of 21 minutes from Waterloo to

Paddington (see Figure 3.11), whereas the simulator adopts a travel time of 14 minutes from Waterloo to Paddington (based on the schedules[6] adopted by Transport for London). Therefore, the two itineraries respectively departing at 10:30 and 10:57 are just the latest departure ones in the context of NRE.



Travel by	Leaving	From	Platform	To	Arriving	Platform	Duration	Additional info
	10:30	London Waterloo [WAT]		London Paddington [PAD]	10:51		0h 21m	
From London Waterloo take the Bakerloo Line (Northbound, Platform 3) which is a direct service to Paddington Underground Station. Check for live travel updates								
	11:06	London Paddington [PAD]		Exeter St David's [EXD]	13:32		2h 26m	
Great Western Railway service from London Paddington to Plymouth + show calling points								

Routes, availability and fares are subject to these provisions

Figure 3.11 The parameter choice of NRE for the Waterloo – Paddington – Exeter route (Source: NRE, accessed 23 Jan 2017)

In fact, the latest departure rule is widely adopted in practice to ensure the conciseness of the computed results (Bast, 2010). This rule is no problem in most cases, but can be problematic in certain scenarios. In order to better understand the potential problem resulting from the latest departure rule, an explanation of the mechanism of minimum transfer time (refer to Subsection 2.3.2) is necessary.

In the terminology adopted by Britain's rail industry, the term minimum transfer time (MTT) is usually called 'connection time' or 'minimum connection time', representing the length of time that must elapse between the advertised arrival time of a feeder train and the advertised departure time of the connecting train within a railway station. That is, the connection between two trains is officially valid only if it satisfies the constraint of the corresponding minimum connection time[7]. Here, in this thesis, the term minimum transfer time (MTT) is adopted to comply with the terminology in Algorithm Engineering. Figure 3.12 gives a more concrete example of some of the MTTs adopted by British railways.

[6] https://tfl.gov.uk/plan-a-journey/

[7] http://www.brtimes.com/#!info?type=conn

London		
Blackfriars	TL	3
Cannon Street	NR	4
Charing Cross	NR	4
City Thameslink	TL	3
Euston	NR	15
Farringdon	LT	3
Fenchurch Street	NR	7
Kings Cross	NR	15
Liverpool Street	NR	15
London Bridge	NR	4
Marylebone	CH	10
Moorgate	LT	
Paddington	NR	15
St Pancras International	NR	15
Victoria	NR	15
Waterloo	NR	15
Waterloo East	SE	4

Figure 3.12 The connection times assigned to London railway stations (Source: Network Rail (2016b)) [NOTE: the middle column lists the corresponding ATOC Code, and the third column lists the corresponding connection times in minutes]

From Figure 3.12, we can see that the MTT assigned to Paddington is 15 minutes, and it is just the difference between the advertised arrival time of the 10:30 tube service (i.e. 10:51) and the advertised departure time of the 11:06 Great Western Railway service in Figure 3.11 (which confirms the latest departure rule). MTTs are generally station-specific, but exceptions exist (although not common) and these exceptions are specially assigned by operators (e.g. the MTT required for Southampton Central is 5 minutes, but 4 minutes is applied to the Bournemouth – Southampton Central – Brighton route, see Figure 3.8).

The incorporation of MTTs into journey planning algorithms is to better model the reality and to ensure that passengers have time to change from one train to another. For dense metropolitan areas, inter-stop MTTs are also assigned between pairs of nearby stops. These intra- or inter-station MTTs, however, are often a trade-off between robust transferring (with generous reserve) and total journey time.

Figure 3.13 gives an illustration of how MTTs work in a journey planning algorithm to construct recommended itineraries for a given query.

In Figure 3.13(a), T1 is a feeder train and its scheduled arrival time at the transfer station is t_{a1}. T2 and T3 are two potential connecting trains that belong to different lines but both call at a given target station (i.e. the railway station at which the traveller transfers to another train or another mode of transport, not necessarily the terminating station of a train line), and their scheduled departure times from the transfer station are t_{d2} and t_{d3}, respectively. The assigned

MTT for the transfer station guarantees that any potential connecting train with a scheduled departure time less than the earliest-allowable departure time t_{ed} (and larger than t_{a1}) could not be chosen as a leg of the recommended itinerary. For those connecting trains with scheduled departure times larger than t_{ed} (e.g. T2 and T3 in this example), however, a journey planning algorithm always 'greedily' selects the one with the earliest scheduled departure time (e.g. T2 in the Figure), regardless of how small the difference between t_{ed} and t_{d2} and how small the difference between t_{d2} and t_{d3}. The similar mechanism holds for the case in which a connecting train has a set of candidate feeder trains (see Figure 3.13(b)): the one with the latest scheduled arrival time (T2 in the example) is 'greedily' chosen (see the previously described examples of Waterloo – Exeter and Bournemouth – Brighton to better understand the mechanism).

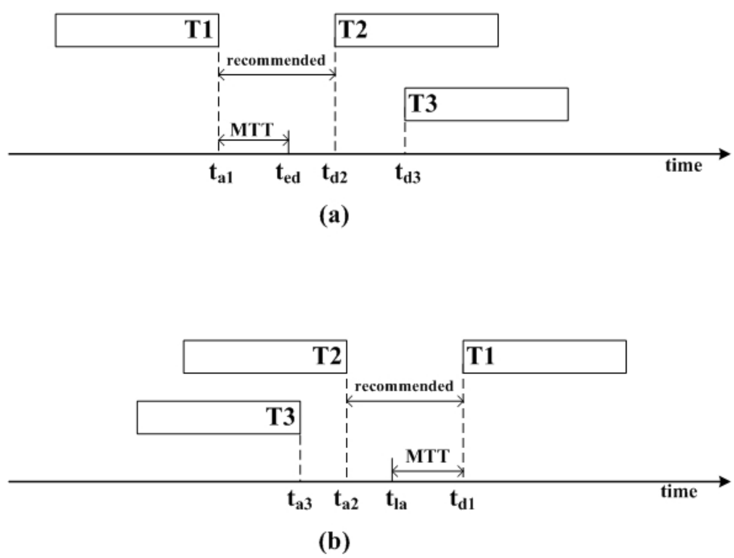


Figure 3.13 An illustration of how Minimum Transfer Times (MTTs) function in 'greedy' journey planning algorithms

3.5 Critical Transfers, Critical Itineraries, and Critical Routes

3.5.1 Introduction

Due to the limitations of existing journey planning algorithms (as have been extensively illustrated in the previous section), some of those transfer-involved itineraries recommended by a journey planning system tend to be sensitive to train delays and cancellations and hence may negatively influence rail passengers' experience of the quality of train services. More specifically, the interplay between MTT, the criterion of earliest arrival, and the mechanism of latest departure would result in tight transfers that may be adopted to construct the recommended itinerary list for a given transfer-involved route. And if the consequence of missed transfers is significant for a particular transfer-involved route, then an improvement of the pre-trip information about (i.e. the recommended itinerary list for) this route should be considered as an option to improve passengers' experience of punctuality and reliability. But how to determine which of those transfer-involved routes are problematic in terms of pre-trip information? How to exploit algorithmic approach to quickly screen out those problematic transfer-involved routes? To answer such questions, several novel concepts should firstly be introduced to make the problem mathematically operable.

3.5.2 Critical transfers

At first glance, the set of recommended itineraries containing delay-sensitive transfers should be quite large due to the fact that there are millions of feasible journeys per day within a national-level railway network and small delays are a common phenomenon that every train is possible to encounter.

In the domain of pre-trip information, however, recommended itineraries involving delay-sensitive transfers are not that common due to the mechanism of MTTs. In reality, although MTTs can cover most transfer scenarios, they tend to be insufficient for certain scenarios in which these insufficiencies together with the 'greedy' mechanism of journey planning algorithms render the recommended transfers prone to delays and cancellations. Figure 3.14 illustrates such delay-sensitive transfer scenarios that MTTs cannot cover.

In Figure 3.14, T1 and T2 are a pair of feeder and connecting trains that satisfies the constraint of the corresponding MTT and appears in a recommended itinerary. That is, this pair of trains simultaneously satisfies the constraint of the corresponding MTT (having been omitted in the figure to reduce distraction) and the rule of earliest arrival (or latest departure). As illustrated in Figure 3.14(a), this transfer becomes delay-sensitive if the scheduled departure time (t_{d2}) of T2 lies within the *most uncertain interval* (MUI). MUI can be imagined as the interval of possible small delays (deviations) of the arrival of T1 (e.g. [-1, 5] mins). Since small delays are quite common and each value in MUI is highly likely to occur in a given trip, the probability of a missed transfer becomes non-negligible under this scenario.

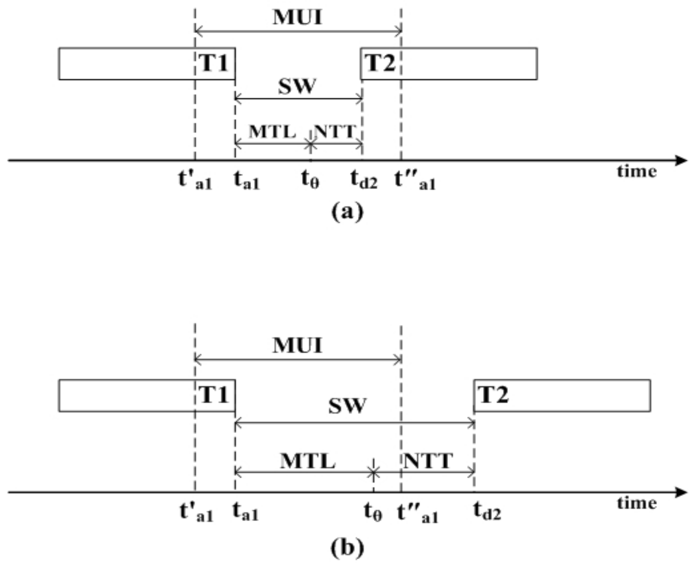


Figure 3.14 An illustration of those delay-prone transfer scenarios that current journey planning systems cannot cover

For scenarios in which t_{d2} lies outside MUI, the transfer is also likely to be delay-sensitive if the *net transfer time* (NTT) is large and the threshold (i.e. t_θ in Figure 3.14) derived from the *maximum tolerable lateness* (MTL) lies within MUI. As illustrated in Figure 3.14, the *scheduled window* (SW) can be decomposed into NTT and MTL: NTT is the physically possible minimum time required to walk from T1 to T2 within the station, and MTL is the size of the maximum buffer for potential delays of T1.

Please note that NTT and MTT are not the same: NTT pertains to a specific pair of feeder and connecting trains, whereas MTT is station-specific and takes into account the NTTs under various scenarios within a given station. Normally, the MTT for a given station is no less than the maximum of all possible NTTs within the station; meanwhile, it is not significantly larger than the maximum of NTTs in case it significantly extends the journey time (reducing the attractiveness of the recommended itineraries).

Although in Figure 3.14(b) SW is relatively large and t_{d2} lies outside MUI, the threshold for the arrival time of T1 (i.e. t_θ) lies within MUI. Combined with the scenario in Figure 3.14(a), those recommended transfers with t_θ lying within MUI can be said to be delay-sensitive.

Although the above transfer scenarios are delay-sensitive and may have a non-negligible risk of missed transfers, their impact on passengers' experience would be limited if, for example, there are a number of alternative transfers at the station or the connecting leg has high-frequency services. Only if the following conditions are simultaneously satisfied, do the corresponding transfer scenarios become problematic and worth to be paid attention to:

(1) The transfer is planned to happen within a given railway station.
(2) The scheduled window (SW) of the transfer is the smallest among all available transfer options with SW $\geq$ MTT.
(3) t_θ lies within MUI (see Figure 3.14).
(4) The service frequency of the connecting leg is low and the length of the connecting leg is long.
(5) The transfer scenario repeats itself periodically (e.g. every weekday) based on a long-term timetable.

Condition (1) reduces the search space to intra-station transfers only. That is, inter-station transfers between nearby stations (e.g. transfers between London terminals) are not taken into account. Adding this restriction is due to the fact that inter-station transfers often involve additional modes of transport (e.g. long-distance walking, bus, underground, etc.) and involve road networks, which would render the estimation of the net transfer time (NTT, see Figure 3.14) between the feeder train and the connecting train difficult.

Condition (2) is to guarantee that it is this (problematic) transfer (rather than others) that is adopted (by existing journey planning systems) to construct a recommended itinerary under the latest departure rule (see Figure 3.13). After all, if a delay-sensitive transfer has been filtered out by journey planning algorithms, there is no need to worry about it in terms of pre-trip information.

Condition (3) has been explained in previous paragraphs. Generally speaking, if the scheduled window (SW) is 'small enough' (as Figure 3.14 illustrates), the influence of uncertainty on journey planning would become non-negligible and hence providing additional information about the potential risk would be meaningful. Conversely, if the scheduled time window is 'large enough', the impact of uncertainty on the connection would be insignificant and the normal timetable-based information could be seen as reliable under most circumstances. But what is the threshold between ‘small’ and ‘large’? The answer is ‘it depends’. It depends on the size of MTL (see Figure 3.14). Since SW can be easily calculated from the timetable, the estimation of net transfer time (NTT) becomes the key, which is further dependent on the distance between the two involved trains within the station, the layout of the station, the familiarity of the traveller with the station, etc. (for a detailed study of the various factors influencing NTT, it is recommended to refer to Guo (2008)). Further details about the considerations in the estimation of NTT (for a given connection) can be found later in the introduction to the screening algorithm.

Condition (4) considers the potential consequence of a missed transfer: if the connecting leg has low frequency and the two end vertices (stations) are geographically far-apart, the potential consequence will be non-negligible and needs to be tackled.

Condition (5) guarantees that the transfer scenario is a long-term existence rather than a short-term noise (e.g. short-term timetables during public holidays, engineering works, etc.). Since the proposed methodology to deal with those problematic transfer scenarios (to be presented in the next chapter) is historical-data-based, the focus is hence not on solving temporary problems but on tackling long-term problems. In fact, current technologies of timetabling and pre-trip information have been able to effectively deal with those *predictable* short-term perturbations such as public holidays, engineering works, etc. (as has been explained in Subsection 2.5.2). And the focus of this thesis is mainly on dealing with those small delays and operational cancellations. Further details can be found later in the next chapter.

A recommended transfer plan (by a journey planning system) that satisfies all of the above five conditions is called a *critical transfer*. Critical transfers are difficult to resolve in current journey planning systems, due to the limitations of existing journey planning algorithms.

3.5.3 Critical itineraries and critical routes

Although critical transfers are problematic in terms of pre-trip information, the negative effect of them would be limited if there are direct alternatives (see the Waterloo – Exeter example in Figure 3.4) or these critical transfers are sparsely distributed on the dimension of time (i.e. the probability that a passenger adopts exactly the problematic transfer would be low).

If all involved transfers in a recommended itinerary are critical transfers, the itinerary is called a *critical itinerary*. If the set of critical itineraries between a given pair of railway stations are densely and evenly distributed in a day (and repeat themselves during a long period of time such as six months), then the corresponding route is called a *critical route*. A *one-transfer critical route* is a critical route that contains exactly one transfer (more precisely, one generic transfer, c.f. Section 3.3). And a *k-transfers critical route* is a critical route composed of exactly k transfers.

From the above definitions, it can be inferred that a critical route (if it is existent in a studied railway system) would be problematic due to the fact that most/all of the itineraries in the recommended list would be delay-sensitive. In the next section, an efficient algorithm is designed and presented, which is able to determine whether there exist critical routes in a given journey planning system and which routes are critical (if they do exist).

Before going to the next section, a clarification needs to be made to distinguish between the notion of 'critical routes' (proposed here) and the notion of 'critical points' in the literature. Andersson et al. (2013) proposes a methodology to identify the robustness weaknesses in a timetable, and these weak points are named *critical points*. Despite some similarity in terminology, there is fundamental difference: the identification of 'critical points' is supposed to happen at the timetabling phase (i.e. before the long-term timetable has been created and finalised), whilst the identification of 'critical routes' is supposed to happen at the operational phase (i.e. after the long-term timetable has been published for passenger information).

3.6 An efficient algorithm to enumerate all critical routes in a railway network

3.6.1 Central idea

The central idea behind the screening algorithm is that instead of scanning the large set of all feasible journeys to identify critical routes, the computational burden can be significantly reduced by firstly screening out all critical transfers and the corresponding one-transfer critical routes (the building blocks), and then permutating the small set of one-transfer critical routes to obtain the list of all critical routes in a given railway network.

Note that the screening algorithm adopts CSA (Connection Scan Algorithm, see Subsection 2.3.4 for reference) as a sub-procedure to simulate an online journey planning system (note: the simulator used in Figure 3.5 is also based on CSA). This choice is, however, not compulsory but largely for convenience. Graph-based journey planning algorithms (e.g. Time-Dependent Dijkstra and Time-Expanded Dijkstra, refer to Section 2.3) often require a great many computational resources to do heavy-preprocessing in order to achieve desirable response times; however, graph-based algorithms have better extensibility and can better support multi-modal journey planning. By contrast, post-Dijkstra algorithms like CSA or RAPTOR (see Subsection 2.3.4 for reference) are array-based and lightweight, which consume fewer computing resources but are mainly designed for public transport networks. Overall, each kind of journey planning algorithms has its own pros and cons, and the reason why CSA is adopted here involves a balance between the consumption of computational resources and the requirement for response times. An algorithmic-level explanation of CSA is to be presented in the next subsection. After that, the screening algorithm proposed is to be detailed.

3.6.2 Connection Scan Algorithm (CSA)

The Connection Scan Algorithm (CSA) is firstly proposed by Dibbelt et al. (2013), and has been proven to be one of the most efficient journey planning algorithms (until now) for timetable-based public transport systems (e.g. rail) (Bast et al., 2015; Wagner, 2015). To better understand the technicalities of the screening algorithm (in the next subsection), it would be helpful to give a brief illustration of CSA.

Below is the pseudo code of the basic version of CSA (i.e. an earliest arrival query, see the subsection 2.3.5 for reference). Before going to the technical details of this algorithm, it is necessary to clarify the meaning of 'connection' in the context of CSA. As mentioned previously in Section 2.5 and the subsection 3.4.2, the term 'connection' is mainly used to describe the interaction between two different trains within a station (e.g. passenger transfers, crew transfers, etc) in the terminology of railway timetabling and operations. Here, in the context of the algorithm, ‘connection’ is a rather abstract notion and is mainly used to refer to a train movement from one station to another. And more precisely, a ‘connection’ in CSA represents an ‘elementary connection’ (refer to the subsection 2.3.2) in a given timetable.

Algorithm 1: CSA (Connection Scan Algorithm)

Input: *Stations*, *Connections*, S_1, S_2, t_1

Output: *Itinerary_recommend*

```
1   // initialising auxiliary arrays
2   for all S_i in Stations:
3       In_connection[S_i] = NULL
4       Earliest_arrival[S_i] = ∞
5
6   // main loop
7   for all C_i in Connections:
8       if t_d(C_i) > Earliest_arrival[S_d(C_i)] and t_a(C_i) < Earliest_arrival[S_a(C_i)]:
9           Earliest_arrival[S_a(C_i)] = t_a(C_i)
10          In_connection[S_a(C_i)] = C_i
11
12  // constructing the recommended itinerary
13  Itinerary_recommend = NULL
14  C_i = In_connection[S_2]
15  while C_i is not NULL:
16          Itinerary_recommend.append(C_i)
17          C_i = In_connection[S_d(C_i)]
18  Itinerary_recommend.reverse()
19  return Itinerary_recommend
```

Recall (in the subsection 2.3.2) that a timetable can be abstractly modelled as a 4-tuple (S, Z, C, D), where S is a set of stations, Z is a set of trains, C is a set of *elementary connections*, and D is a set of service dates. In this 4-tuple, an *elementary connection* in set C is itself a 5-tuple (Z_i, S_d, S_a, t_d, t_a), which can be interpreted as follows: a train Z_i departs the current stop station S_d at time t_d and arrives at the immediately next (scheduled) stop station S_a at time t_a. The mechanism of CSA is just built on such an abstraction of a master timetable.

In Algorithm 1 (CSA), a master timetable is firstly reformatted and stored into two arrays: *Stations* (i.e. all active stations in a railway network) and *Connections* (i.e. all elementary connections in the timetable). CSA then receives a time query (S_1, S_2, t_1) (note:S_1 is the source station, S_2 is the target station, and t_1 is the desired departure time from S_1; refer to the subsection 2.3.5) and returns a recommended (earliest arrival) itinerary. In an execution of CSA, two auxiliary arrays are firstly initialised (Lines 1 – 4): *In_connection* stores all the incoming (elementary) connections for each station, and *Earliest_arrival* stores the earliest arrival time for each station. The second stage is the main loop of the algorithm: the array *Connections* is fully scanned to obtain the earliest arrival time at the target station (i.e. S_2) and mark all the involved stops en route. Then, in the final stage, a post-processing procedure is run to construct and return the recommended itinerary. Note that $S_d(\cdot)$, $S_a(\cdot)$, $t_d(\cdot)$, and $t_a(\cdot)$ in the above algorithm respectively represent the departure station, the arrival station, the (scheduled) departure time, the (scheduled) arrival time of a given (elementary) connection C_i.

3.6.3 Critical Routes Finder (CRF): the screening algorithm

Algorithm 2 below presents the pseudo code of the developed screening algorithm (called Critical Routes Finder) for identifying and enumerating all the critical routes (defined as in Section 3.5) within a given railway system. The algorithm (i.e. CRF) involves a number of sub-procedures (including the aforementioned CSA-based journey planning simulator), a lot of data cleaning and processing, and several carefully designed heuristics to accelerate the executions. A Python implementation of CRF is presented in Appendix B, the source code of which is composed of approximately 1500 lines (of commands). Therefore, rather than being viewed as one algorithm, CRF can be more appropriately described as a set of several interdependent algorithms.

CRF is generally composed of five major steps. All the notations in italics are one-dimensional list (array) objects, those in bold are two-dimensional tables, and uppercase letters are constant parameters. The only exception is CSA in Step 4, which is short for Connection Scan Algorithm (as previously described) and is not a parameter but a procedure.

Algorithm 2: CRF (Critical Routes Finder)

Input: a long-term timetable that contains information about stations, lines, trips, stop times, calendar, and minimum transfer times

Output: a list of all critical routes in the studied railway system

```
// Step 1: determine the set of all transfer stations in the network
for each in Lines:
   record seq and store it into StopSequences
for each pair in StopSequences:
   if no shared origin and destination:
      if not inverse to each other:
         compute intersec
         if len(intersec) == 1:
```

```
            store intersec into TransferStations

// Step 2: construct station-view timetables for transfer stations
for each station i in TransferStations:
   if MinimumTransferTime(i) > UPPER:
      continue
   extract from StopTimes the records pertaining to i and store into a separate table Table_i
   sort Table_i by scheduled arrival time
   merge Table_i with Stations, Calendar, etc. to introduce additional columns for scanning
   store Table_i into StationViewTimetables

// Step 3: scan StationViewTimetables to obtain a candidate list of critical transfers
for each table j in StationViewTimetables:
   delete those records with service days < DAYS
   flag those records with line headway > HEADWAY and store into Connecting
   assign LOWER_j and UPPER_j for scanning
   for each record k in table j:
      for each record m with dep(m) in [arr(k)+LOWER_j, arr(k)+UPPER_j]:
         if line(m) in Connecting and dist(station(j), destination(m)) > DIST:
            if diff(platform(k), platform(m)) > DIFF:
               store (origin(k), station(j), destination(m)) into CandidateList

// Step 4: double-check CandidateList to obtain the list of critical transfers and one-
//           transfer critical routes
extract from StopTimes the timetable for a normal service day
for each pair of origin and destination in CandidateList:
   run a multi-criteria CSA on the timetable to obtain a list of recommended itineraries
   if the recommended itineraries follow exactly one route with exactly one transfer:
      store the recommended route into RecommendedList
intersect CandidateList with RecommendedList to obtain CriticalTransfers
drop duplicates in CriticalTransfers to obtain 1-Transfer-Routes

// Step 5: permute 1-Transfer-Routes to obtain the list of all critical routes
k = 2
while k < K:
   enumerate k-permutations of 1-Transfer-Routes and store them into CandidateList_k
   double-check CandidateList_k to obtain the final list of k-Transfers-Routes (repeat Step 4)
   store k-Transfers-Routes into CriticalRoutes
   k = k+1
   if len(k-Transfers-Routes) == 0:
      store 1-Transfer-Routes into CriticalRoutes
      return CriticalRoutes
      terminate
```

Step 1 and Step 2 can be seen as pre-processing steps. These two steps can significantly reduce the search space and the computational burden on scanning tables. This is because realistic railway systems are often sparse networks in which only a small subset of all railway stations are potential transfer stations. Relevant symbols are as follows:

- *seq* means the stop sequence of a given train line.
- *intersec* is the intersection set of two line-specific stop sequences.
- origin/destination means the originating/terminating station of a given line.
- UPPER is the upper bound for an insufficient MTT.
- **StopTimes** is a table that stores all the scheduled arrival, departure, and passing times at all station stops for all lines within a given rail network.
- **Stations** is a table that stores station-related information about e.g. name, location, special identifier in a given code system, etc.
- **Calendar** is a table that specifies the operational and non-operational dates for each train line within a given timetable period.
- *StationViewTimetables* means a list of (line-specific) timetables grouped by station.

Step 3 and Step 4 are the core part of CRF. While Step 3 is mainly to check Conditions (3), (4) and (5) in the four conditions for critical transfers, Step 4 is mainly to check Condition (2). Note that Condition (1) has been implicitly taken into account in Steps 1 and 2. Step 3 involves several network-specific parameters, and the considerations behind parameters selection are to be explained later in the application of CRF to the National Rail timetable currently used by Britain's passenger rail system. Relevant symbols are as follows:

- DAYS is the threshold for the number of operating days within a timetable period.
- HEADWAY is the threshold between low-frequency and high-frequency services.
- *Connecting* is the candidate list of connecting legs.
- LOWER_j and UPPER_j are station-specific parameters that bound the interval [MTT, UPPER].
- dep(m) and arr(k) are the scheduled departure time of m and the scheduled arrival time of k.
- DIST is the threshold between near and far in terms of geographical distance between the transfer station and the destination station (of a given connecting line).

- DIFF is the difference between the platform number of the feeder train and that of the connecting train.

Step 5 introduces two stopping conditions. One is to stop the algorithm when no new critical routes enter the result set (which is very natural). The other is a constraint of maximum number of transfers (i.e. the parameter K), which is to accelerate the termination under the extreme case in which there are critical routes involving unrealistically large number of transfers.

The whole algorithm has been carefully implemented and tested in the analysis of the National Rail timetable currently in use in Britain's passenger rail system, the execution of which is proven to be quite efficient (up to 3 mins in total). Since all the involved parameters are network- or station-specific, the choice of each parameter is to be detailed in the analysis.

CRF (Critical Routes Finder) has been developed to locate those critical routes (defined in Sections 3.4 and 3.5) within a large search space (composed of millions of possible pairs of source and target stations). The creation and adoption of this particular approach has been mainly based on the consideration that it would be much more efficient than a Brute-Force approach (i.e. firstly enumerate all possible routes between all possible pairs of source and target stations, and then check all these routes one by one).

3.7 An analysis of British National Rail timetable using CRF

3.7.1 Introduction

Britain has one of the busiest railways in Europe with about 22,500 trains running every day and 1.7 billion rail journeys made per year (Network Rail, 2016b; ORR, 2016). Since passenger rail journeys (in Britain) have more than doubled over the last two decades (ORR, 2016), the infrastructure capacity utilisation also increases, reaching its limit at critical parts (Network Rail, 2016a). A higher capacity utilisation tends to bring more frequent delays (Olsson and Haugland, 2004), and train delays and cancellations are currently quite common in British railway system.

On the other hand, rail passengers in Britain tend to rely increasingly on web-based information sources to plan their journeys, especially when planning unfamiliar and/or long-distance journeys (Farag and Lyons, 2008). In the following, the CRF algorithm presented in the previous section is to be applied to the current National Rail timetable (i.e. the published long-term timetable) adopted by British railways to identify those weak points (i.e. critical routes) in the pre-trip timetable information (i.e. those recommended itineraries by NRE).

3.7.2 Data preparation

In this particular analysis, three sets of relevant data are prepared: the National Rail timetable data, the London Underground timetable data, and historical train movements data about arrival and departure delays at major stations.

Although generally stable, the long-term (planned) timetable of National Rail is updated every six months. In this analysis, the latest version (at the time of writing up this thesis) is adopted, which is valid from 11 December 2016 to 20 May 2017. Although different formats are available: PDF (Network Rail, 2016b), XML (ATOC, 2016), and GTFS (http://www.gbrail.info/), a dataset of GTFS format is adopted because GTFS data are well-organised and easier to process. The GTFS timetable is updated every week to reflect minor modifications to rail operations in the following week, and the exact file adopted is the one published on 19 November 2016.

The London Underground timetable[8] is also involved. Recall that in Steps 4 and 5 in the screening algorithm (i.e. CRF), a CSA-based journey planning simulator is run to check the candidate list of critical transfers and critical routes. Since many journeys across Britain involve inter-station transfers between London Terminals (e.g. Waterloo, Victoria, etc.), London Underground is often a good choice to complete these inter-station transfers.

The historical delay data collected are a 12-months dataset that contains information about a huge amount of recorded arrival and departure events (logs) at a number of major railway stations. The observation period is from 14 Sept 2015 to 13 Sept 2016, which crosses three

[8] https://tfl.gov.uk/travel-information/timetables/

timetable periods: 17 May 2015 – 12 Dec 2015, 13 Dec 2015 – 14 May 2016, and 15 May 2016 – 10 Dec 2016. The records are organised by station: the investigated stations are Bournemouth, Southampton Central, Brighton, Exeter St Davids, Cardiff Central, Birmingham New Street, Clapham Junction, Leeds, Doncaster, Manchester Piccadilly, Edinburgh, Liverpool Lime Street, Sheffield and Preston (14 stations in total). The relevant data records have been downloaded and stored into separate files every day (during the 12-months observation period) from Realtime Trains (RTT). The reason why adopting RTT data is that RTT data are generally well-structured and easier to process than those poorly-structured raw data from Network Rail. Although RTT is not an official source of historical rail data, it is a well-known data consumer of Network Rail's data feeds (rather than a data creator). Note that since the database of RTT has limited storage space, the historical data are renewed on a weekly basis (i.e. old data are removed to leave space for new data). Hence, RTT data needs to be carefully and timely collected, before the relevant records are removed.

3.7.3 Parameters selection

Step 2 (of the screening algorithm CRF) involves an important parameter: the upper bound for potentially insufficient minimum transfer time (MTT). Recall that in the introduction to critical transfers and critical routes (Section 3.5), the mechanisms of MTT and MUI (most uncertain interval) have been respectively explained, but the relationship between them has not been clarified. This is because although MTT and MUI are inter-related, both of them are network-specific (i.e. may vary among different countries) and their relationship is largely indefinite. Here, considerations are explained about how to determine MUI and hence determine the threshold for potentially insufficient MTT in the context of British railways.

The statistics shown in Figure 3.15 below are calculated from the 12-months historical data (Figure 3.15(a) and (b)) and the National Rail timetable data (Figure 3.15(c)).

Figure 3.15 (a) and (b) respectively present the distribution of arrival and departure delays at the 14 studied railway stations in Britain (refer to the previous introduction in Subsection 3.7.2) during the 12-months observation period. The total number of effective observations (i.e. null values and cancelled trains are not included) is 1,405,785 for arrival events, and 1,439,873 for departure events. The observed arrival delays have 283 distinct values (unit: minutes) with the minimum and maximum being -104 and 436, respectively. The observed departure delays have 287 distinct values, with the extreme values being -83 and 781, respectively.

Δarr	percentage
0	19.55
-1	16.79
1	13.53
2	9.41
-2	7.50
3	6.55
4	4.58
-3	3.80
5	3.21
6	2.30
-4	1.91
7	1.66
8	1.22
9	0.93
-5	0.90
10	0.73
[-5, 10]	94.57
[1, MAX]	48.74
[3, MAX]	25.80
[5, MAX]	14.66
N = 1,405,785	
MIN = -104, MAX = 436	

(a)

Δdep	percentage
0	42.45
-1	15.46
1	13.32
2	8.40
3	5.47
4	3.54
5	2.46
6	1.72
7	1.23
8	0.91
9	0.71
10	0.56
[-1, 10]	96.22
[MIN, 0]	58.14
N = 1,439,873	
MIN = -83, MAX = 781	

(b)

MTT	percentage
5	74.97
4	12.69
6	3.52
3	3.02
2	2.36
7	0.93
1	0.89
10	0.77
8	0.35
15	0.31
9	0.08
30	0.04
20	0.04
12	0.04
[MIN, 5]	93.93
N = 2585	
MIN = 1, MAX = 30	

(c)

Figure 3.15 Statistics for arrival delays, departure delays, and minimum transfer times in British railways

Generally, the arrival delays have a 'flatter' distribution than the departure delays (c.f. Figure 3.15), indicating that the uncertainty in arrival events tends to be larger than that in departure events. We can also see from both distributions that small variations account for the vast majority of the total. Although the sample adopted is quite large in terms of the number of observations, it only accounts for a small portion of the whole network and not necessarily representative: these statistics should not be seen as the exact probabilities (e.g. the size of arrival delays may be systematically underestimated using this sample, for rail operators usually take measures to improve punctuality at major stations but allow larger delays at small stations). But one thing is clear from these statistics: small arrival delays are not that rare, and they can result in delay-sensitive transfers if combined with relatively punctual departures and insufficient MTTs.

Figure 3.15 (c) presents the distribution of MTTs for all British railway stations. The statistics are calculated from the National Rail timetable data (in GTFS format). We can see that among the 2585 stations, around 94% are assigned a MTT no more than 5 minutes. This is not surprising because most of the stations in Britain's passenger rail system are not major transfer stations and the net transfer time (NTT, refer to Figure 3.14 in Subsection 3.5.2) within a small station is trivial. But there is a possibility that some transfers happen at small stations but there are not enough reserves to offset the impact of prevalent small delays. For those major transfer stations with large MTT, there is also a possibility that some transfers require large NTTs and there are not enough reserves for small delays.

So how to determine the upper bound for MTT? The key is firstly determining an upper bound for MUI. Recall that a transfer plan is considered as valid only if the scheduled window between the advertised arrival and departure times of two trains is greater than a predefined minimum transfer time (i.e. SW ≥ MTT, see Figure 3.13 in Subsection 3.4.2). Meanwhile, the scheduled window between two trains can be seen as the sum of the net transfer time and the maximum tolerable lateness (i.e. SW = NTT + MTL, see Figure 3.14 in Subsection 3.5.2), and the threshold for the arrival of the feeder train should lie within MUI (i.e. $t_\theta \leq t''_{a1}$, see Figure 3.14 in Subsection 3.5.2) were it recognised as a critical transfer (refer to subsection 3.5.2). Therefore, the upper bound for MTT is dependent upon the upper bound for MUI: $MTT \leq SW = NTT + (t_\theta - t_{a1}) \leq NTT + t''_{a1} - t_{a1}$.

From Figure 3.15 (a) we can see that the arrival delays in the interval [-5, 10] account for about 95% of the total, and hence 10 (mins) can be set as the upper bound for MUI. This choice may be questioned because the percentage of 10 is only 0.73%, which seems not that uncertain. But considering that these are aggregated statistics without differentiating between regional and long-distance trains and the critical routes we aim to find out (if existent, as defined in Subsection 3.5.3) often involve long-distance trains, this choice should be appropriate. More importantly, since a series of further screenings are to be executed at later stages (of CRF), we only need to obtain a rough estimation of a network-level MUI at this stage, and adopting a wider MUI could reduce the error of omitting some important (but delay-sensitive) transfers. Based on the observation that the majority of intra-station transfers in National Rail can be completed within 3 minutes (i.e. the maximum of NTT is around 3 minutes across the network), the parameter UPPER can hence be set to 12 minutes (the maximum MTT no more than (10+3) is 12, see Figure 3.15 (c)). That is, we need only to scan the timetables for those transfer stations with assigned MTTs no more than 12 minutes, and those transfer stations with MTTs larger than 12 are not possible to cause critical transfers.

The other parameters (in CRF) to choose are involved in Step 3. The parameter DAYS is to filter out those short-term noises: only long-existence transfers are taken into account. Since one timetable period is about six months in Britain, this parameter is set to 180 (d) in this analysis. The parameter HEADWAY is to identify those low-frequency train lines: it is set to 30 minutes (i.e. two services per hour), which is in line with most British rail operators' delay compensation policies (e.g. Virgin Trains[9]). With respect to the two station-specific parameters LOWER_j and UPPER_j, the choices are based on the following considerations: LOWER_j is always set to MTT_j because only those pairs of trains with SW (scheduled window) larger than MTT are likely to enter the set of recommended itineraries; UPPER_j is set to 10 (i.e. the upper bound for MUI) if MTT_j < 10, and is set to 12 (i.e. the upper bound for MTT) if 10 <= MTT_j <= 12 (see Figure 3.14 for illustration). The parameter DIST is to guarantee that the length of a connecting leg is long enough and hence the consequence of a missed transfer is difficult to offset by shifting to local public transport (e.g. bus, tram, etc.). Considering the specific characteristics of British public transport, this parameter is set to 40 kilometres in this analysis. The parameter DIFF takes into account the correlation between a pair of feeder and connecting trains: the delays of the two involved trains tend to be positively correlated if the two trains are allocated to the same platform; the farther apart they are, the lower the potential correlation between them and hence the more likely the transfer is delay-sensitive. And DIFF is also used to filter out those pairs of trains with small NTTs (net transfer time, see Figure 3.14).

Table 3.1 The parameters adopted in this analysis

UPPER (mins)	12
DAYS (d)	180
HEADWAY (mins)	30
LOWER_j and UPPER_j (mins)	LOWER_j = MTT_j & UPPER_j = 10 if MTT_j < 10 LOWER_j = MTT_j & UPPER_j = 12 if 10 ≤ MTT_j ≤ 12
DIST (km)	40
DIFF	DIFF ≥ 1 if MTT_j ≤ 5 DIFF ≥ 2 if 5 < MTT_j < 10 DIFF ≥ 3 if 10 ≤ MTT_j ≤ 12

[9] https://www.virgintrains.co.uk/delayrepay

For the convenience of reference, the assigned values to all the involved parameters in this analysis are summarised in Table 3.1. It should be noted that these values are not compulsory: they can be adjusted as necessary.

3.7.4 The screening results

In the following, the screening results (i.e. critical routes) as well as the intermediate results in each step are to be presented. Moreover, the execution time (i.e. computational time) for each step is also recorded to enable the knowledge about the screening algorithm's (i.e. CRF's) performance in terms of efficiency. The code is written in Python 2.7 (refer to Appendix B) and run on a machine with Intel® Core™ i7-4700MQ CPU, 2.4 GHz, and 8 GB of RAM.

By adopting UPPER = 12, the two pre-processing steps (Step 1 and Step 2) reduce the search space from the set of 2585 stations to a small subset of 277 stations. The computational time for these two steps is around 34 seconds.

After the execution of Step 3, a candidate list of 379 potential critical transfers across British railways is obtained. The computational time for Step 3 is around 75 seconds (based on the parameters presented in Table 3.1).

Step 4 is to check each of the transfers in the candidate list to see whether it is realistic. This is because those transfers in the candidate list are only critical in theory and there may be many unrealistic scenarios such as detours. After filtering out those apparently unrealistic transfers (i.e. the distance between origin and target stations less than 20 km), the number of transfers in the candidate list is reduced to 248. For those inconspicuous detours, the CSA-based journey planning simulator (see Appendix B) is employed to complete the filtration task. The optimisation criteria adopted are scheduled journey time and number of transfers. The timetable adopted is a full-day timetable (including the London Underground timetable) for a normal working day during the studied timetable period (here, 25 Jan 2017 is adopted). After the check-up of the 248 transfers in the candidate list, a final list of 13 critical transfers and their corresponding one-transfer critical routes are identified:

Ebbw Vale Town – Cardiff Central – Nottingham;
Knottingley – Wakefield Kirkgate – Nottingham;
Liverpool Lime Street – Manchester Piccadilly – Doncaster;
New Mills Central – Manchester Piccadilly – Scarborough;
London Kings Cross – York – Scarborough;
Weymouth – Southampton Central – Brighton;
Harwich Town – Manningtree – Norwich;
Sudbury (Suffolk) – Marks Tey – London Liverpool Street;
Marlow – Maidenhead – Oxford;
Rugeley Trent Valley – Birmingham New Street – Hereford;
Hoxton – Clapham Junction – Alton;
Kirkby (Merseyside) – Manchester Victoria – Huddersfield;
Oxford – Reading – Gatwick Airport.

The computational time for this step is about 51 seconds. Please note that this step can be accelerated by further optimising the implementation of the journey planning simulator: since the adopted implementation is in pure Python (normally an order of magnitude slower than a C++ counterpart), more efficient implementation can be adopted if the candidate list is large (e.g. thousands of transfers).

Step 5 in this analysis converges (terminates) very quickly: no such case exists that the ending point of one route is the starting point of another (called 2-permutations in the algorithm), let alone k-permutations ($k \geq 2$). Therefore, a lot of checking and rechecking is saved and the computational time for this step is trivial (< 1s).

Summing up the five steps, the screening of a full list of critical routes in British railways can be completed within 3 minutes (about 160s), which is quite efficient considering the large search space for the whole network. Please note that this list is based on the planned timetable for the period from 11 December 2016 to 20 May 2017, and is subject to the changes in the long-term timetable. Note also that critical routes may contain critical sub-routes. That is, some of the intermediate stops (stations) along a given critical route may themselves construct *child routes* following the same transfer pattern with their *parent route* (i.e. the identified critical route) and the child routes also satisfy the definitions of critical transfers and critical routes. Although critical sub-routes are not common in reality due to the mechanism of current journey planning algorithms (e.g. Bournemouth – Southampton Central – Brighton and Ebbw Vale Town – Cardiff Central – Birmingham New Street are two identified critical sub-routes in the above list), the existence of critical sub-routes makes the set of critical itineraries and the number of passengers influenced often larger than the estimations based solely on those parent routes.

3.8 A further investigation into the train delay data of British railways

3.8.1 Introduction

In the application of the developed screening algorithm (i.e. CRF) to British National Rail timetable (in the previous section), a large dataset containing historical train delay information during a 12-months period between 14 Sept 2015 and 13 Sept 2016 has been briefly described and been exploited to determine a network-specific parameter (i.e. the upper bound for insufficient minimum transfer time) for British railways (c.f. Subsection 3.7.3). Although the main objective of this chapter (i.e. introducing and explaining the concept of critical routes and identifying them in Britain's passenger rail system) has been achieved up to this point, a more detailed analysis of the collected historical delay data may help better understand passenger train delays in British railways. In fact, few previous studies have utilised big data to investigate train delay distributions in a national-level railway network, and scientific knowledge of train delay distributions remains fragmented and limited. Hence, this section is mainly aimed at integrating the existing empirical evidence in the literature and generating updated knowledge about passenger train delays.

3.8.2 Statistical modelling and the results

Figure 3.15 (i.e. (a) and (b)) has presented some of the delay statistics of the recorded 1,405,785 arrival events and 1,439,873 departure events happening at the 14 studied railway stations (c.f. Subsection 3.7.2) during the 12-months period between 14 Sept 2015 and 13 Sept 2016. But the whole picture of the observed arrival and departure delays has not been shown. Hence, the whole distribution of arrival delays and that of the departure delays are firstly displayed in Figure 3.16 and Figure 3.17 below, respectively.

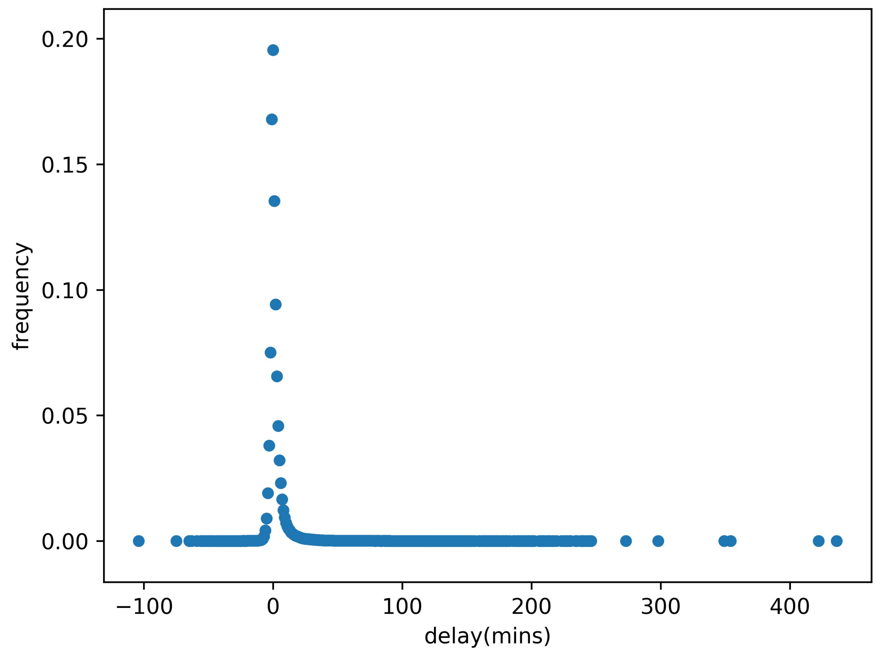


Figure 3.16 The distribution of arrival delays in British railways (based on a large sample between Sept 2015 and Sept 2016)

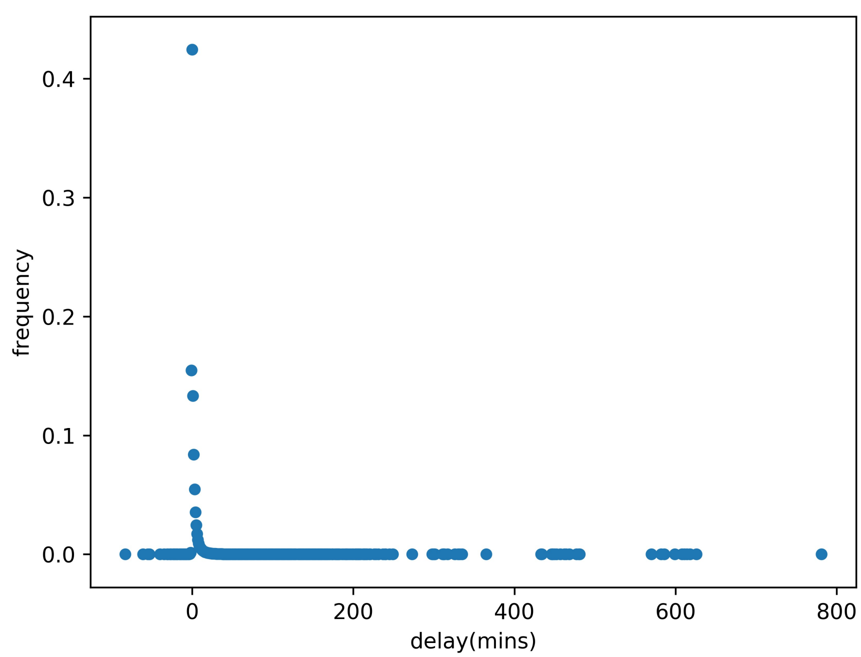


Figure 3.17 The distribution of departure delays in British railways (based on a large sample between Sept 2015 and Sept 2016)

From Figure 3.16 we can see that the overall distribution is asymmetric: those positive delays (i.e. those on the right hand side of zero) tend to decay at a slower rate than those negative delays (i.e. those on the left hand side of zero), and those positive delays are characterised by a quite long tail (up to 436 mins delay, c.f. Figure 3.15(a)). This asymmetry is even more obvious in the distribution of departure delays shown in Figure 3.17: the left tail (corresponding to those negative delays) is significantly shorter than the right tail (corresponding to those positive delays), and the decay rate of those negative delays are much faster than that of those positive delays and also faster than that of those negative delays in Figure 3.16.

Based on Figures 3.16 and 3.17 and the empirical results in several previous studies (e.g. Yuan, 2006; Briggs and Beck, 2007), it can be inferred that the whole distribution (incorporating both negative and non-negative delays) of arrival/departure delays is most likely to be a compound/mixed distribution of a number of random variables, which cannot be described by a simplistic statistical model. Therefore, a separate investigation may be needed.

Since negative delays (i.e. early arrivals or departures) are widely regarded as ‘on time’ in the rail industry, previous relevant studies (Yuan, 2006; Briggs and Beck, 2007; Bergström and Krüger, 2013) have mainly focused on the modelling of positive delays. To maintain consistency and facilitate the analysis, the focus of this section is also placed on those positive delays.

Based on the above considerations, four candidate statistical models have been developed (the fitted curves are depicted in Figure 3.18 below), which respectively correspond to the following four functional forms: q-exponential, power law with exponential cutoff, lognormal, and Weibull.

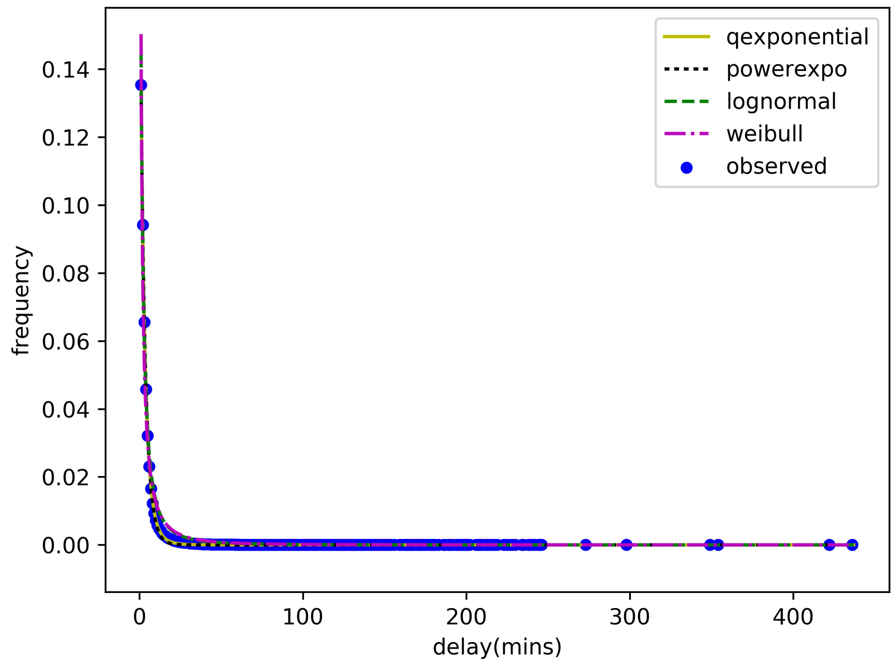


Figure 3.18 Four candidate statistical models developed to fit the (positive) arrival delay data

The reason why these four functional forms have been chosen (for comparison) is mainly based on the consideration that they have been respectively recommended by previous relevant studies. Briggs and Beck (2007) utilised a large British dataset containing over two million train departures for the period Sept 2005 – Oct 2006 to model (positive) departure delays in British (passenger) railways, and they found that the sample data can be 'accurately described' by the so-called 'q-exponential functions' (which can be viewed as a compound/mixed distribution of a number of random variables). Bergström and Krüger (2013) adopted a large Swedish dataset containing over three million valid train arrivals for the two-year period of 2008 and 2009 (1.6 million for each) to model positive arrival delays, the results of which indicate that the exponential distribution can be used to describe those extreme values in the tail and the power law with an exponential cutoff (i.e. a combination of the power law and the exponential distribution) may be used to model the overall distribution of positive arrival delays. Yuan (2006) conducted a comprehensive statistical analysis of the train traffic data recorded at The Hague HS station in the Netherlands during the whole month of September 1999 (approximately 10,000 trains recorded). The empirical results from Yuan (2006) generally favour the Weibull distribution and the lognormal distribution as the best-fit statistical models of train delays.

To compare the capabilities of the four candidate functions in describing the historical train delay data adopted in this section, the obtained best-fit models respectively corresponding to the four functions (the specific forms and parameters are to be detailed later in Table 3.2) are presented (depicted) in Figure 3.18. It can be seen from Figure 3.18 that all of the four candidate models can generally fit the (positive) arrival delay data quite well, and their performances are indistinguishable on such a scale. But if we 'magnify' the granularity of the y-axis (i.e. the frequency axis), their differences become identifiable. Figure 3.19 below subdivides the range of the observed arrival delays (corresponding to the x-axis in Figure 3.18) into four sub-intervals (i.e. (0, 15], (15, 60], (60, 120], and (120, 436]), and respectively compares the observed delays with the four fitted curves on each sub-interval adopting different granularities (corresponding to the y-axes in Figure 3.19) to reflect local details.

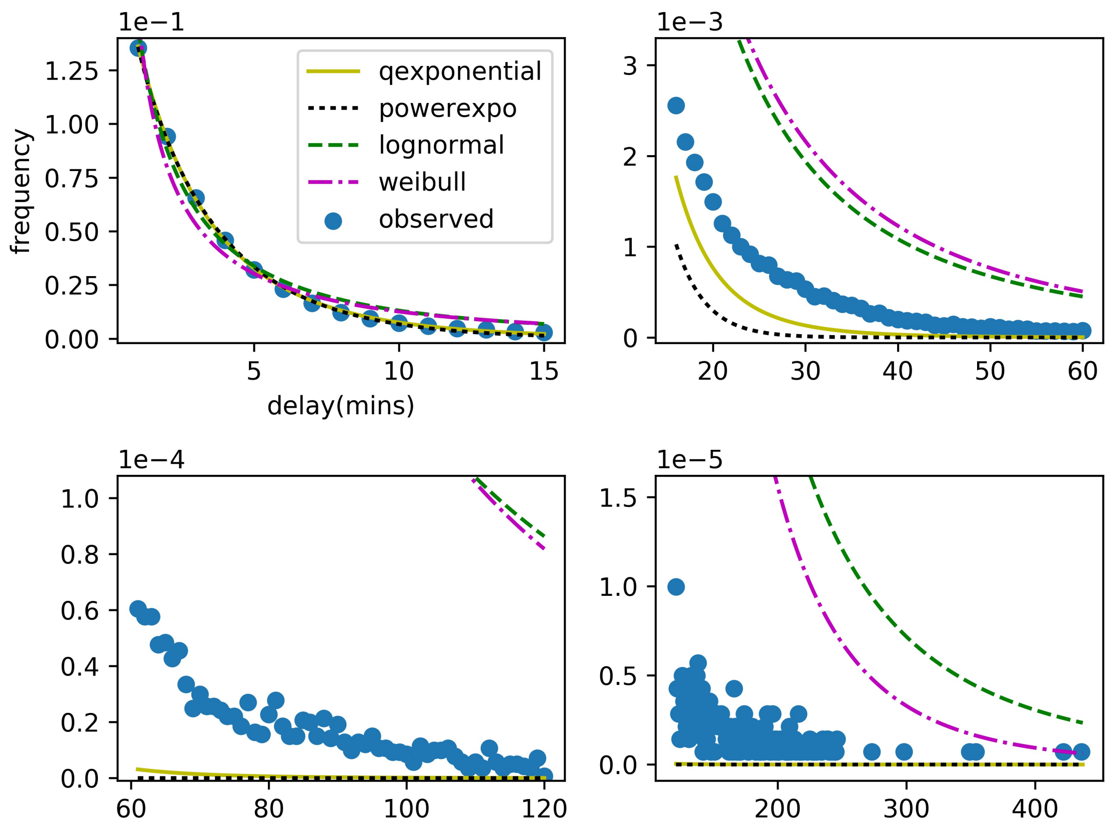


Figure 3.19 Comparisons between the four candidate models for (positive) arrival delays

From Figure 3.19 we can see that q-exponential and power law with cutoff can generally better fit the data (i.e. smaller deviations from the observations) than Weibull and lognormal on the sub-interval of (0, 15]. Moreover, it can also be seen from the figure that with the increase of x values (i.e. delay size), q-exponential and power law with cutoff tend to systematically underestimate the observed values (i.e. frequency) while Weibull and lognormal tend to systematically overestimate the observed. However, these deviations (from the observations) should not be over-interpreted (based on the graphical descriptions in the figure) due to the fine granularity adopted.

With respect to the overall performance of each model in describing the delay data, it can be speculated from the graphical description in Figure 3.19 that q-exponential and power law with exponential cutoff tend to be more able (than the other two) to describe the data on the whole domain (i.e. from 1 to 436 on the x-axes). To examine this speculation, a quantitative index – mean absolute error (MAE) – is respectively calculated for each candidate model. Mathematically, MAE is defined by the following equation:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} | f_i - y_i | = \frac{1}{n}\sum_{i=1}^{n} | e_i | \qquad \text{Eq. (0)}$$

In the above equation, y_i corresponds to the observed value at the i^{th} position/point on the domain, f_i corresponds to the predicted/theoretical value (based on a specific model) at the i^{th} position/point on the domain, and $|e_i|$ represents the absolute error at the i^{th} position/point on the domain. The reason why choosing MAE as the index for comparison is mainly based on the consideration that it has a generic definition and is not constrained by some specific functional form: each of the four candidate models developed is the best-fit one among those of the same functional form, and MAE provides a straightforward way to measure goodness of fit and make cross-functional comparisons.

Table 3.2 The specific parameters and indices of the four candidate models for (positive) arrival delays

candidate model	PDF (probability density function)	best-fit parameters	MAE (mean absolute error)
log-normal	$\frac{1}{x\sigma\sqrt{2\pi}}\exp\left[-\frac{(\ln x-\mu)^2}{2\sigma^2}\right]$	μ=1.048, σ=2.212	6.6×10^{-4}
Weibull	$\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$	k=0.436, λ=2.400	7.8×10^{-4}
power law with cutoff	$cx^{-\alpha}e^{-\lambda x}$	α=0.116, λ=0.304, c=0.184	1.6×10^{-4}
q-exponential	$c(1+b(q-1)x)^{1/(1-q)}$	q=1.132, b=0.413, c=0.203	1.1×10^{-4}

Table 3.2 presents the specific functional forms and the obtained parameters of the four candidate models that have been depicted in Figures 3.18 and 3.19. All of the relevant computations in curve fitting have been conducted using Python 2.7, with the aid of several statistical packages/libraries such as NumPy, SciPy, etc. Moreover, the corresponding MAEs to the four models are also presented in the rightmost column. The obtained results of MAEs in the table confirm our initial speculation from Figure 3.19: q-exponential and power law with cutoff generally outperform the other two models in terms of the overall performance in describing the data (the specific results are to be further interpreted in the next subsection).

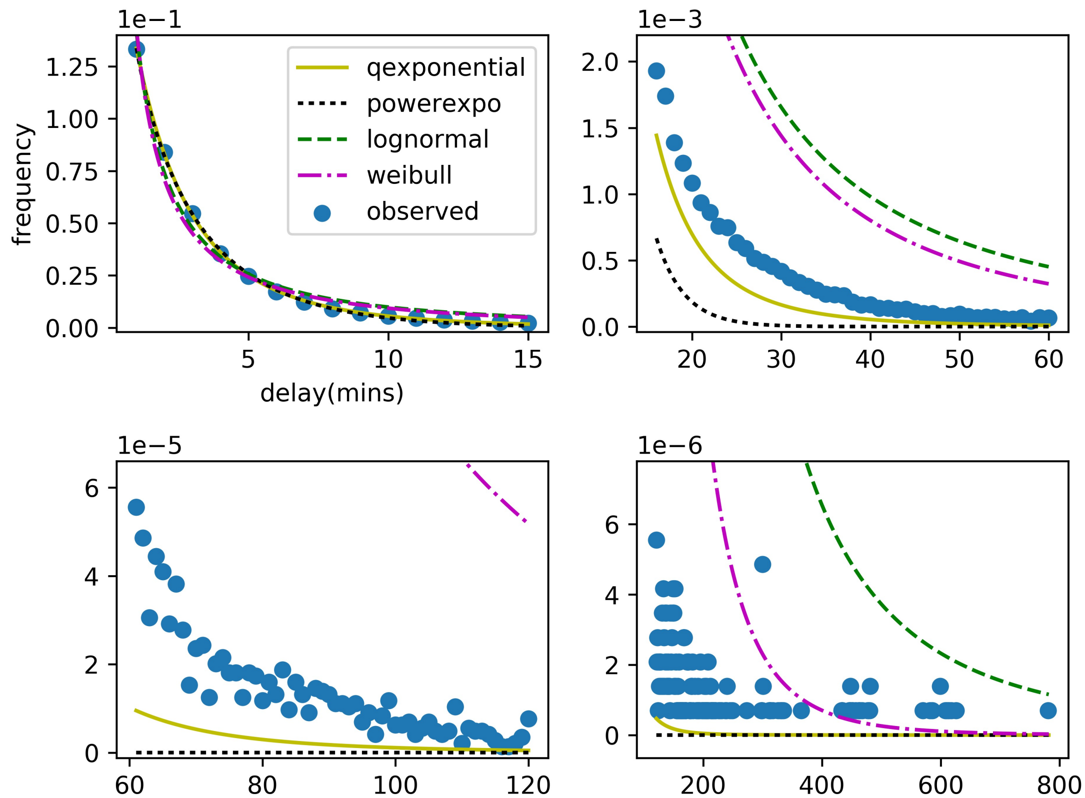


Figure 3.20 Comparisons between the four candidate models for (positive) departure delays

In the above, several candidate statistical models for (positive) arrival delays in British railways have been developed and compared. A similar statistical analysis has also been conducted of those (positive) departure delays in British railways (c.f. Figure 3.17). The graphical descriptions and the specific parameters and indices (MAEs) of the developed models are presented in Figure 3.20 and Table 3.3, respectively. Comparing Figure 3.20 and Table 3.3 with their counterparts above (i.e. Figure 3.19 and Table 3.2), we can find some similarities: despite the difference in specific parameters and indices, q-exponential and power law with cutoff generally better fit the recorded delay data.

Table 3.3 The specific parameters and indices of the four candidate models for (positive) departure delays

candidate model	PDF (probability density function)	best-fit parameters	MAE (mean absolute error)
log-normal	$\frac{1}{x\sigma\sqrt{2\pi}}\exp\left[-\frac{(\ln x-\mu)^2}{2\sigma^2}\right]$	μ=0.588, σ=2.690	5.3×10^{-4}
Weibull	$\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}e^{-(x/\lambda)^k}$	k=0.392, λ=1.279	4.7×10^{-4}
power law with cutoff	$cx^{-\alpha}e^{-\lambda x}$	α=0.266, λ=0.304, c=0.181	1.3×10^{-4}
q-exponential	$c(1+b(q-1)x)^{1/(1-q)}$	q=1.207, b=0.558, c=0.227	6.6×10^{-5}

3.8.3 Interpretation

In this subsection, the graphical and numerical results presented in the previous subsection are to be further interpreted by linking them with operational practices and with previous relevant studies.

Firstly, the differences between the overall distribution of arrival delays and that of departure delays (c.f. Figure 3.16 and Figure 3.17 in the previous subsection) are not difficult to understand. At least the following three underlying forces may have resulted in the differences between the two distributions. Firstly, the asymmetry between the recorded arrival events and the recorded departure events (in the sample) is an identifiable factor. Theoretically, each arrival event would correspond to at least one departure event (and vice versa) in the universal set of all arrivals and departures (from the perspective of cause and effect). However, the sample data adopted in this section is only a subset (of all arrivals and departures in the studied railway network) containing records of 14 medium-to-large-sized stations (c.f. Subsection 3.7.2) despite its large sample size. That is, those departure delays in the sample (c.f. Figure 3.17) may not be totally attributable to those arrival delays in the sample (c.f. Figure 3.16), and vice versa. Secondly, some operational routines may also have resulted in the differences between the arrival and departure delay distributions. For example, those negative delays in departure records are significantly less than those in arrival records (c.f. Figures 3.15, 3.16, and 3.17), which may be attributable to an operational practice that prohibits early departures (c.f. Goverde, 2005). Thirdly, some timetabling techniques may be another influencing factor. For example, running time supplements/allowances (c.f. Subsection 2.5.2) are often added in the timetable design process, which may have contributed to the relatively large proportion of early arrivals (c.f. Figures 3.15 and 3.16).

With respect to the results presented in Figures 3.18 ~ 3.20 and Tables 3.2 ~ 3.3, they tend to indicate that all of the four statistical models built could generally describe the sample data of train delays in British railways quite well (with a mean absolute error of a magnitude of 10^{-4}) and hence could be utilised to make delay estimations/predictions. Moreover, these results generally corroborate some of the findings/claims in previous relevant studies. Both Yuan (2006) and Bergström and Krüger (2013) have mentioned that the overall distribution of (positive) train delays is likely to be a compound/mixed distribution of a number of random variables, which can be largely confirmed by the empirical results presented in this section: the two compound distributions (i.e. q-exponential and power law with cutoff) do outperform the other two 'pure' distributions (i.e. Weibull and lognormal) in terms of goodness of fit. And the main finding of Briggs and Beck (2007) – q-exponential functions can 'accurately' describe the distribution of train delays in British railways – can also be corroborated by the results presented in the previous subsection: q-exponential has the least MAE (mean absolute error) among the competitors in both Table 3.2 and Table 3.3.

Although some interesting information can be extracted from the analysis of the four statistical models presented in the previous subsection, these findings ought to be treated with caution. An interesting and relevant question is raised here: how accurate can be regarded as 'accurate' (quoted from Briggs and Beck (2007)) when using statistical models to describe train delay distributions? The answer is likely to be 'it depends'. For those small delays with high probabilities (e.g. (0, 15], c.f. Figures 3.19 and 3.20), all of the candidate models may be treated as quite accurate in estimating/predicting delay probabilities due to a far lower magnitude of errors/deviations (i.e. 10^{-4}) than the corresponding delay probabilities themselves (i.e. $10^{-2} \sim 10^{-1}$). In contrast, even the most 'accurate' model (i.e. q-exponential) may not be regarded as accurate enough for those heavy delays with low probabilities (e.g. > 60, c.f. Figures 3.19 and 3.20), for the errors/deviations (with a magnitude no less than 10^{-5}) would exceed the corresponding delay probabilities themselves (with a magnitude no greater than 10^{-5}). Therefore, it is suggested (based on the empirical results presented in this section) that further (separate) analyses of the distribution of those heavy/large delays be conducted in future research.

Despite the fact that the statistical analyses conducted in this section are not specific to a particular route, station, or season, some interesting findings can still be drawn at the network level.

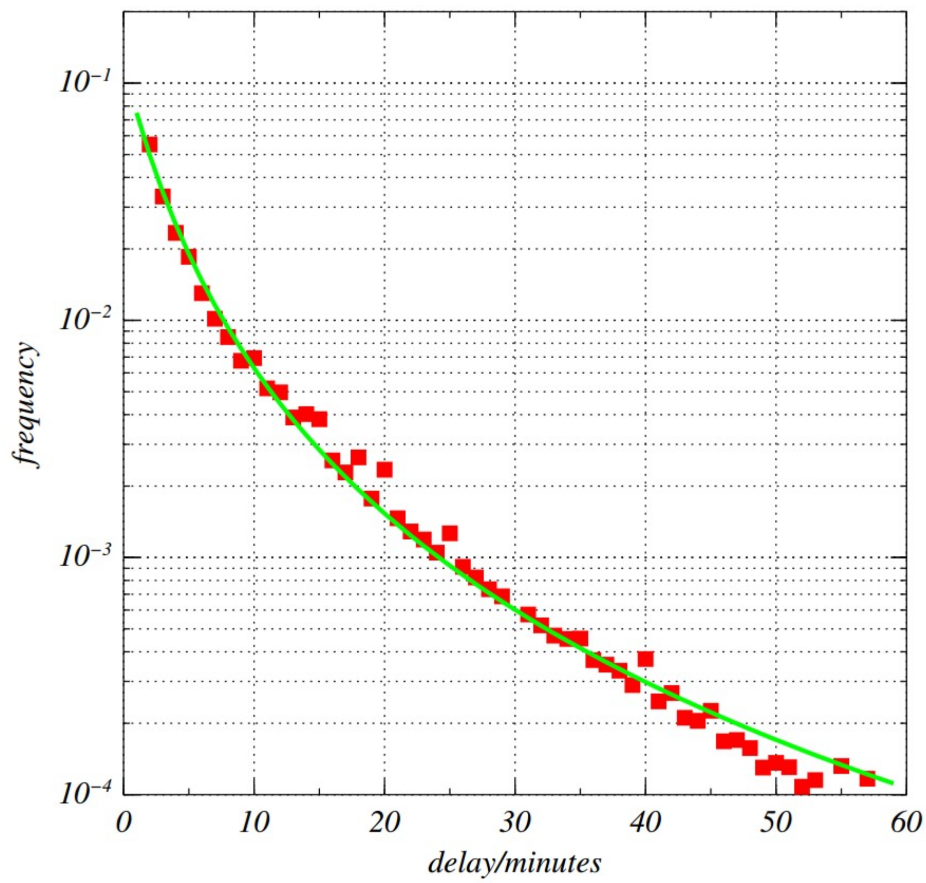


Figure 3.21 The best-fit (q-exponential) curve for the departure delay data of British railways between Sept 2005 and Oct 2006 (Source: Briggs and Beck, 2007)

As previously mentioned, Briggs and Beck (2007) utilised big data to model train delays in British railways and they have built a q-exponential model to fit the collected data on departure delays for 23 railway stations[10] between September 2005 and October 2006. The best-fit parameters have been $q = 1.355 \pm 8.8 \times 10^{-5}$ and $b = 0.524 \pm 2.5 \times 10^{-8}$ for their

[10] These include Bath Spa, Birmingham, Cambridge, Canterbury East, Canterbury West, City Thameslink, Colchester, Coventry, Doncaster, Edinburgh, Ely, Ipswich, Leeds, Leicester, Manchester Piccadilly, Newcastle, Nottingham, Oxford, Peterborough, Reading, Sheffield, Swindon, and York.

developed model. Although the normalisation parameter of c (see Table 3.2 or Table 3.3 for the specific functional form of q-exponential) has not been explicitly presented in their paper, it can be inferred from the graphical description of their developed model (see Figure 3.21) that the parameter c is roughly around 0.12.

Once the best-fit q-exponential for the period September 2005 to October 2006 has been restored, we can then investigate the change in the distribution of departure delays in British railways over the last decade by comparing the model developed by Briggs and Beck ($q=1.355$, $b=0.524$, $c=0.12$) with the q-exponential model developed in this section ($q=1.207$, $b=0.558$, $c=0.227$; c.f. Table 3.3). Figure 3.22 below provides a graphical comparison of the two models. Note that the long tails (of the two models) have not been depicted based on the consideration that the trend of each of the two fitted distributions can be easily derived from the figure (i.e. the decay rate of those triangles is faster than that of those circles). As has been shown in this section and in Briggs and Beck (2017), both of the two q-exponential models depicted in Figure 3.22 can generate quite accurate estimations of train delays for their corresponding observation periods (i.e. Sept 05 to Oct 06 and Sept 15 to Sept 16), especially for those small delays at the head part of the distributions.

It can be seen from Figure 3.22 that those small delays (up to 8) have happened more frequently between Sept 2015 and Sept 2016 while those medium-to-large-sized delays (greater than 8) have happened less frequently between Sept 2015 and Sept 2016. This implies that the main focus of rail operators in Britain may have been placed on the management of those medium-to-large-sized delays (based on the assumption that both of the two q-exponential functions can 'accurately' describe those medium-to-large-sized delays), which could directly benefit rail operators (e.g. less fines and/or more subsidies) by the improvement of those existing performance measures such as PPM (Public Performance Measure, c.f. Network Rail, 2017). Setting aside the potentially little attention paid to those small delays (by rail operators), a possible reason for the increase in small delays may lie in the (rapid) growth in rail usage over the last decade (and the resulting crowdedness in stations and trains) (DfT, 2016b and 2017). Although the increase in small delays might not be a severe problem for those transfer-free journeys, it may lead to an increasing risk of broken connections for those transfer-involved journeys, especially for those Critical Routes (c.f. Section 3.5). Hence, apart from those medium-to-large-sized delays, rail operators in Britain should pay additional attention to those small delays to improve the overall quality of rail services.

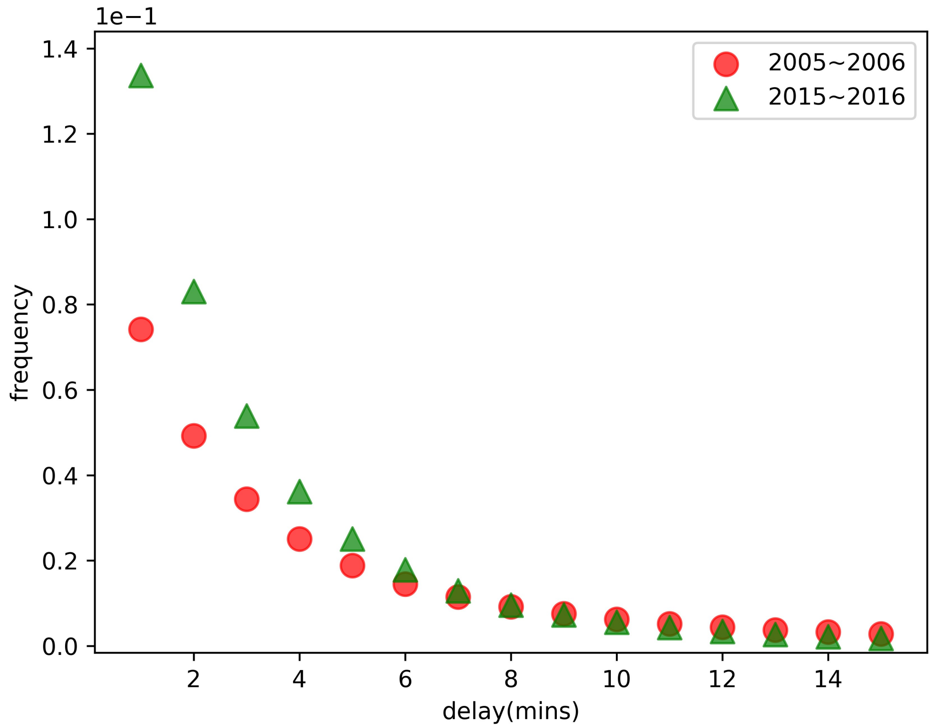


Figure 3.22 The change in the departure delay distribution (for British railways) over the past decade

3.9 Conclusions

Transfer-involved rail routes receive relatively less attention from rail operators compared with direct routes, not only in terms of timetable design but also in terms of pre-trip passenger information. As an increasingly popular information tool, online journey planning systems such as National Rail Enquiries (and its mobile version) in Britain play an important role in the pre-planning of transfer-involved routes. However, the quality of the pre-trip information about those transfer-involved routes (i.e. the recommended itinerary list) is often disregarded, due to the limitations of existing journey planning technologies. At the algorithmic level, these limitations are embodied by the interaction between several competing forces (i.e. earliest arrival, latest departure, and minimum transfer time). Since these limitations come from within the framework of existing journey planning algorithms itself, they are difficult to be overcome unless a breakthrough can be made to jump out of the existing framework (which seems an unachievable goal in the foreseeable future).

An alternative solution is to identify those weak points (i.e. problematic transfer-involved routes) under the existing algorithmic framework, and then focus on tackling this small subset of problematic routes. In order to automatically and efficiently identify those problematic routes in terms of pre-trip journey planning, it is necessary to introduce some novel concepts to make the screening problem mathematically operable. These introduced concepts are: critical transfers, critical itineraries, and critical routes. Roughly speaking, a critical itinerary is composed of critical transfers, each of which is delay-sensitive and is associated with high consequence if missed. And if the recommended itinerary list (by a journey planning system) is full of critical itineraries, the corresponding route would be problematic in terms of punctuality and reliability and is called a critical route.

An efficient screening algorithm, named Critical Routes Finder (CRF), is developed and implemented to check whether there exist critical routes within a given railway system and to find out, if existent, which of those transfer-involved routes are critical. The screening algorithm is then applied to analyse the current National Rail timetable adopted by British railways to identify those critical routes within Britain's passenger rail system. The performance of the screening algorithm is promising in terms of computational efficiency. The screening results show that more attention should be paid to such transfer-involved routes as London Kings Cross – York – Scarborough, Weymouth – Southampton Central – Brighton, etc to improve the pre-trip information about these routes.

A statistical analysis of a large sample of train delay data has also been conducted for British railways for the period September 2015 to September 2016. The empirical results tend to indicate that all of the four studied candidate functions (i.e. lognormal, Weibull, power law with cutoff, and q-exponential) can generate quite accurate predictions of those small-sized delays, but none of them give a desirable performance in fitting those medium-to-large-sized delays. Overall, q-exponential outperforms the other three candidate functions in terms of goodness-of-fit. Comparing the latest version of q-exponential (derived from the 2015/16 data) with a previous version of q-exponential (derived from a 2005/06 sample), a non-negligible increase in small-sized delays has been identified in British railways, which implies that a better management of those small delays may be necessary to alleviate the potential problem of transfer-involved journeys.

Chapter 4
Tackling Critical Routes: a historical-data-based approach

4.1 Introduction

The existence of critical routes in a passenger rail system would be problematic in terms of pre-trip passenger information. As illustrated in Chapter 3, the recommended itinerary list for a critical route would be full of delay-sensitive transfers, resulting in poor-quality pre-trip information in terms of punctuality and reliability.

These critical routes should, ideally, be resolved in the timetabling (i.e. timetable design) process, which belongs to the tactical planning phase rather than the operational planning phase (c.f. Section 2.5). In reality, however, railway timetabling is a complicated process that involves a delicate balance of technical feasibility, convenience for passengers, and the interests of different operators. For example, if a critical route involves two different rail operators sharing no rolling stock or crew, they may lack the incentive to reschedule those delay-sensitive transfers if the transferring passengers are a minority group or if the rescheduling would increase the operational cost of the other processes.

A more feasible solution to critical routes is improving the quality of the pre-trip information (i.e. those recommended itineraries) about these routes. This chapter hence focuses on finding information-related strategies to cope with those critical routes within a given railway system. Although no previous studies are directly related to or pay special attention to critical routes, a review of the existing solutions to some similar problems is firstly presented in Section 4.2 to help understand the big picture of the state-of-the-art journey planning technologies. After that, the central idea and the technicalities of the proposed (historical-data-based) approach are explained in Section 4.3. Section 4.4 then presents several illustrative examples in the context of British railways to show the potential applications of the proposed approach. Section 4.5 points out and illustrates a potential limitation of the proposed (data-driven) solution to critical routes to stimulate further research in the relevant directions. Section 4.6 concludes this chapter.

4.2 Existing information-related approaches to tackling missed transfers

Missed transfers (connections) have long been a weak point in terms of pre-trip passenger information, despite the rapid development of journey planning technologies. Missing a transfer could be a serious problem, especially for long-distance connections running with low frequency. This section is aimed at presenting a brief review of the state of the art of the various information technologies that have been developed to mitigate the problem of missed transfers. The review covers a variety of sources of references – ranging from mature real-world applications to immature prototypes in the literature to raw algorithmic ideas.

4.2.1 Frequently updating the underlying timetables

Müller-Hannemann and Schnee (2009), Allulli et al. (2014), Cionini et al. (2014), and Delling et al. (2014a) are the advocates of incorporating dynamic (delay) information into static timetable information systems. And a number of real-world pre-trip timetable information systems (e.g. National Rail Enquiries, DB Bahn, etc) have largely implemented this kind of algorithmic solution in recent years.

While Müller-Hannemann and Schnee (2009) and Cionini et al. (2014) are centred on enhancing graph-based algorithms (c.f. Subsection 2.3.3) to enable dynamic updating of the underlying timetables (efficiency), Allulli et al. (2014) and Delling et al. (2014a) focus on investigating to what extent the exploitation of dynamic information (real-time GPS data in the context of their studies) can improve the static timetable information (effectiveness).

Although these studies have obtained generally desirable results, this category of approaches (i.e. frequently updating the underlying timetables) suffers from the same limitation with those real-time delay/disruption alerts (c.f. Sections 3.2 and 3.3) – the accuracy of the dynamic information cannot be guaranteed until it is very near to the time of travel. Therefore, they contribute little to the pre-planning of transfer-involved journeys.

4.2.2 Reliability rating based on simplistic models

Disser et al. (2008) and Schnee (2009) propose an algorithmic approach that computes reliable journeys (itineraries) by multi-criteria optimisation (c.f. Subsection 2.3.7). Delling et al. (2014b) also adopt this method in their proposed RAPTOR algorithm (c.f. Subsections 2.3.4. and 2.3.7). The idea is to add into a given journey planning algorithm a predefined 'reliability rating model' to evaluate how 'reliable' each individual journey plan is and employ 'reliability rating' as an additional criterion to optimise (besides journey time and number of transfers). More specifically, this method is based on two introduced concepts called 'reliability of transfer' and 'reliability rating', respectively. For a given journey plan, a measurement of 'reliability of transfer' is firstly calculated for each involved transfer by a predefined 'reliability rating function': $rel(x) = \mu - e^{\ln(\mu-\theta)-\frac{1}{\alpha}x}$, in which x is the buffer time at the transfer station (defined as the scheduled time window between the feeder and connecting trains minus a predefined station-specific minimum transfer time), and α=8， θ=0.6， μ=0.99 are predefined parameters obtained from empirical evidence (based on German data). Then, after calculating the reliability indices for individual transfers, a 'reliability rating' can be assigned for the (whole) journey plan by multiplying all these reliability indices together.

Two problems arise when looking through this method. First, the reliability indices generated by the 'reliability rating function' are not realistic reliability statistics and hence are difficult to interpret. Second, for a given railway station, transfer events occurring at different times of a day and different days of a week often have the same buffer time (calculated from the above definition) due to the periodicity of train schedules. All of these transfer events would be assigned to the same 'reliability rating' according to the univariate (i.e. the variable 'x' in the function) 'reliability rating function', which is counter-intuitive and seems to have omitted a lot of other realistic factors (e.g. the characteristics of infrastructure and rolling stock, weather, driver behaviour, etc).

4.2.3 Backup information

Goerigk et al. (2013; 2014) transfer some of the notions originating from robust timetabling into timetable information, and propose the notion of 'recoverable robust timetable information'. Their idea is to compute journey plans that maximise the use of 'safe' transfers robust in all (/most) simulated delay scenarios and provide back-up plans to guarantee the arrival at target stations. Dibbelt et al. (2014) takes this one step further: instead of a single path, each individual journey plan is represented as a decision graph composed of all 'good' back-ups at all involved transfer stops (see Figure 4.1 for an illustration). Keyhani (2017) employs more complicated but more realistic stochastic methods (than those simplistic ones) to evaluate and compare reliability, and proposes its own version of pre-trip backup information called *complete connections* – a complete connection comprises a train connection and an associated set of *alternative continuations* to the destination.

The limitation of Goerigk et al. (2013) and Goerigk et al. (2014) lies mainly in the heavy pre-processing spent on enumerating a very large set of possible delay scenarios, which impedes the method's applicability in practice. The limitation of Dibbelt et al. (2014) is twofold: on the one hand, most single-path itineraries generated by existing journey planning systems have been robust enough in most scenarios, and decision graph representations seem too complicated to be useful and may be misleading; on the other hand, the 'delay model' underpinning this method seems too simplistic and suffers from similar limitations of the model adopted in Disser et al. (2008) and Schnee (2009). The limitation of Keyhani (2017) is mainly embodied by its complicated representation of results and its heavy reliance on the Assumption of Independence (which is far from realistic) in conducting the addition/multiplication/convolution operations of multiple random variables.

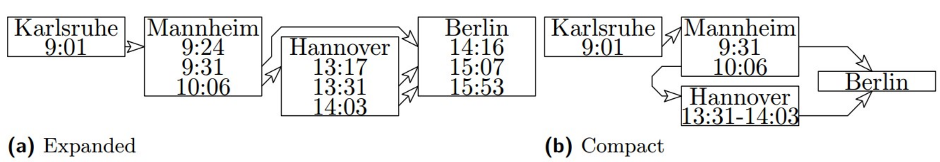


Figure 4.1 An illustrative example of what a 'decision graph' should look like (Source: Dibbelt et al., 2014) Note: a recommended journey plan is no longer a 'single path', but should be represented as a set of back-up plans according to the idea of Dibbelt et al. (2014).

4.2.4 Robust routing based on historical data

Böhmová et al. (2013) and Böhmová et al. (2015) propose a novel algorithm that computes journey plans robust under 'typical' delay scenarios by learning from historical delay data

(Pröger (2016) provides more detailed illustrations and evaluations about this methodology). The algorithm introduces a new form of itinerary representation (i.e. only a recommended route with a departure time; no intermediate arrival/departure times along the route) and computes robust journey plans based on 'recorded timetables' that are constructed by realised stop times (i.e. arrival/departure times and passing times).

Compared with those simplified 'delay models', this method can better reflect the temporal and spatial variations inherent in public transport. However, since this method is designed for high-frequency urban public transport systems (e.g. bus and tram), most of its notions and the associated algorithm cannot be transplanted into intercity or international railway systems (the urban public transport system is often dense enough to provide many different (and similarly attractive) routes between any pair of source and target nodes, but this characteristic is not applicable to the intercity rail system, especially those critical routes within the intercity rail system).

4.2.5 Customisable transfers

A recently developed functionality in real-world pre-trip timetable information systems is called *customisable transfers* (see Fiugres 4.2 and 4.3 for illustrations). As the name implies, customisable transfers means that a rail passenger now could adjust the parameter of MTT (minimum transfer time) and hence directly control the recommended itinerary list. For example, suppose there are now two alternative transfer plans (for a given journey) – one with a scheduled transfer time of 8 minutes, and the other with a scheduled transfer time of 11minutes. If a passenger chooses an MTT of 5 minutes, then the one with 8-minutes transfer time will be recommended. But if the passenger sets the MTT to 10 minutes, then the one with 11-minutes transfer time will be recommended. At the algorithmic level, a modification of the parameter of MTT corresponds to an updating of the list/array storing all MTTs for different stations and connections, the task of which could be efficiently completed using current algorithmic techniques.

Although this functionality could be a practicable way to deal with transfer-related problems, it has two potential limitations. Firstly, it implicitly assumes that a transfer plan with more scheduled transfer time would be more reliable (robust to the impact of delays/disruptions) than another transfer plan with less scheduled transfer time. However, this assumption does not necessarily hold true in some cases, especially when taking into account the diversity and heterogeneity that exist in station size, station layout, the characteristics of stairs, lifts, ramps, etc. Secondly, the functionality of customisable transfers also implicitly assumes that a rail passenger has sufficient experience/knowledge to judge whether a certain MTT can help achieve a good balance between reliability and efficiency. Clearly, this does not necessarily hold true for those occasional/inexperienced rail users. For example, an infrequent user having selected an MTT of *x* minutes on the basis of poor background knowledge may be penalised by not being given information about interchanges designed to be achieved in (*x*-1) minutes, which may become acceptable connections with a time no less than MTT if the arriving (feeder) train is early and/or the departing (connecting) train is late. And even if a passenger is a frequent user of rail transport, he/she may also have difficulty in selecting an 'optimal' MTT (based solely on train schedules), considering the various factors influencing train movements.

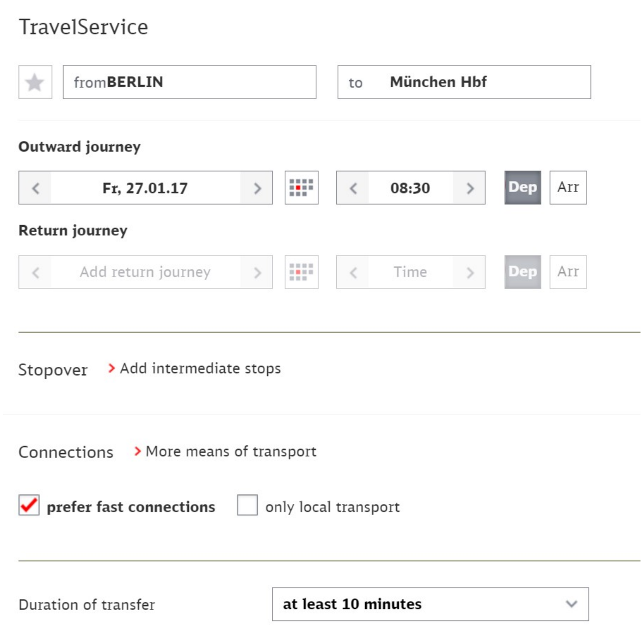


Figure 4.2 Customisable transfers Example One: Deutsche Bahn (Source: www.bahn.com, accessed 25 Jan 2017)

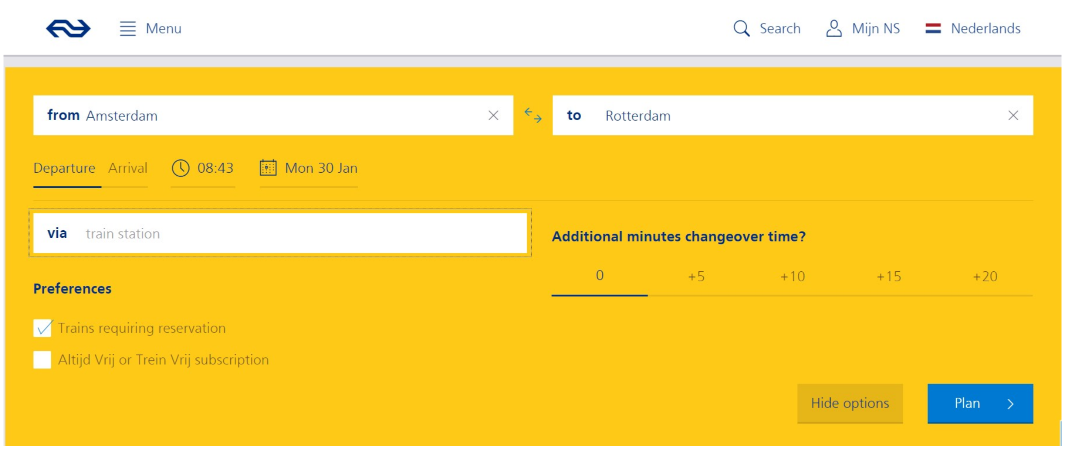


Figure 4.3 Customisable transfers Example Two: NS (Source: www.ns.nl/en, accessed 25 Jan 2017)

4.2.6 Increasing transfer buffers

A simpler approach than the other categories of approaches is simply increasing/adding buffer times for certain (important) transfers (c.f. Pachl, 2014; Pröger, 2016; Caimi et al., 2017; Keyhani, 2017). Such an approach/idea can be implemented in two different ways – either by adding more buffer times into the underlying timetables or by increasing MTTs (Minimum Transfer Times, c.f. Section 3.4) in pre-trip itinerary computation and recommendation.

Adding more buffer times directly into the underlying timetables looks like a radical surgery that would eradicate missed transfers. Its price is, however, considerably high so that it is seldom considered as a good option – such an implementation would not only have an uncontrollable/unpredictable negative impact on capacity utilisation (Figure 4.4 provides an illustration) but also result in a non-negligible reduction in the competitiveness (/attractiveness) of pre-trip itinerary recommendations (especially in the case of critical routes; to be further explained in Subsection 4.2.8). In fact, the study of how to efficiently add and wisely allocate buffer times in timetable design and optimisation has been an active research direction for a while, but a sufficiently satisfying (i.e. simple but powerful) solution/answer has not yet found (c.f. Parbo et al., 2016; Caimi et al., 2017). Considering the limitation of the existing timetabling technology and the long-term growth trend of rail demand (c.f.

Armstrong and Preston, 2017), such an implementation (i.e. adding more buffer times directly into the underlying timetables) would obviously not be a sustainable solution.

Compared with direct operations on the underlying timetables, increasing MTTs in computing and recommending transfer-involved itineraries can be said a light implementation. Although increasing MTTs would hardly erode capacity, it could bring non-negligible reductions in competitiveness (/attractiveness). Empirical evidence in the relevant literature (c.f. Pröger, 2016; Keyhani, 2017) has revealed such non-negligible reductions for the general case of transfer-involved routes. Subsection 4.2.8 is to present illustrations of such non-negligible reductions for the special case of critical routes (which are much more significant than the general case). Due to this non-negligible negative effect on competitiveness (/attractiveness), such a light implementation (i.e. increasing MTTs in computing and recommending transfer-involved itineraries) has seldom been considered as a good solution, either.

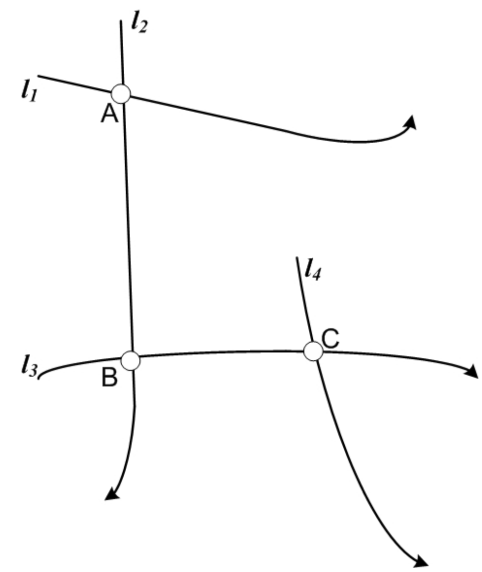


Figure 4.4 An illustration of the potential consequence of adding additional buffer time to a critical route

[Suppose this is a small part of a large railway network and all the irrelevant stations and lines are hidden to reduce distraction. Station A is an intermediate stop of both Line l_1 and Line l_2. The transfer from l_1 to l_2 via A, denoted by $<l_1, \text{A}, l_2>$, is feasible but critical based on the underlying timetable. $<l_2, \text{B}, l_3>$ and $<l_3, \text{C}, l_4>$ are feasible and not critical. Suppose we add e.g. an additional 8-minute transfer buffer to $<l_1, \text{A}, l_2>$ by changing the scheduled departure time of l_2 at A from $Sch_{dep}(l_2, \text{A})$ to $Sch_{dep}(l_2, \text{A}) + 8$. Then, the scheduled departure time of l_2 at B would be postponed and $<l_2, \text{B}, l_3>$ might become critical or even infeasible. Then, we have to modify the schedule of l_3 to fix this new problem of $<l_2, \text{B}, l_3>$. If the modified schedule of l_3 influence the criticality or feasibility of $<l_3, \text{C}, l_4>$, then we have to further modify the schedule of l_4 to resolve the problem... The ultimate result of this domino, an increase in idle capacity induced by added buffer times, would have to be 'digested' by reducing the capacity provision at the relevant lines and stations within or outside this part of the whole network.]

4.2.7 Performance statistics

The use of performance statistics to learn about and control the quality of rail services is not unusual among European railways, and punctuality and reliability are one of the major concerns of European rail operators (c.f. Subsection 2.5.4). In Britain, the industry standard adopted to evaluate and compare punctuality and reliability is called Public Performance Measure (PPM) (see Figure 4.5 for an illustration). PPMs are calculated from several predefined threshold values and are represented by aggregate statistics indicating the network- or subnetwork-level performance in terms of punctuality and reliability. Although these performance statistics are useful in helping rail operators and the government supervise the overall performance of rail services within a certain area during a certain period of time, they tend to be of little help to individual passengers who are more concerned with disaggregated statistics about the performance of the particular lines/routes that they (will) use.

Computing and disseminating disaggregated statistics is technically impracticable in the past due to the limitation of computing resources and the unavailability of detailed data about train movements. In recent years, with a significant development of computer hardware and the increased availability of detailed and open-source rail data, the computation and dissemination of disaggregated statistics is no longer impossible, but a new bottleneck arises of how to extract from huge amounts of train movements data as much useful information as possible (RRUKA, 2015). In this context, several experimental passenger information systems (websites and/or mobile applications) that provide information about disaggregated performance statistics have been emerging in Britain in recent years. Some examples are Recent Train Times (www.recenttraintimes.co.uk/), Fasteroute Delay Explorer (delayexplorer.fasteroute.com/#/), and My Train Journey (www.mytrainjourney.co.uk/). Despite the difference in the representations of disaggregated performance statistics, all of them are driven by the open rail data from Britain's rail industry and their statistics are all oriented toward specific train services. Figures 4.6 and 4.7 illustrate how train-oriented performance statistics are presented in these information systems.

Performance by train operator

Train Operating Company	PPM % period 10, 2015/16	PPM % period 10, 2016/17	PPM Moving annual average (MAA)
Abellio Greater Anglia	90.3	90.0	88.9
Arriva Trains Wales	89.9	88.3	91.5
c2c Rail	95.1	96.1	94.7
Caledonian Sleeper	74.1	87.2	87.8
Chiltern	92.9	93.5	93.4
Crosscountry	88.8	88.9	89.6
East Midlands Trains	94.5	96.2	91.8
First Hull Trains	76.8	80.6	82.8
Transpennine Express	82.8	87.6	88.5
Govia Thameslink Railway	76.0	72.3	74.6

Figure 4.5 An illustration of aggregated performance statistics: PPM in British railways (Source: Network Rail, 2017)

From Figures 4.6 and 4.7, we can catch a glimpse of the major characteristics of these state-of-the-art performance information tools: the statistics are oriented to specific trains and are based on historical train movements data over the last several weeks (i.e. eight weeks in Figure 4.6 and four weeks in Figure 4.7); and like those aggregate statistics in Figure 4.5, these disaggregated statistics are also calculated from several predefined threshold values or industry standards (e.g. 5 mins late, 15 mins late, right time, reliability, etc). Although Figure 4.6 and Figure 4.7 share several important characteristics, we can also see some differences between them. While Figure 4.6 (i.e. Fasteroute Delay Explorer) tends to be generally better at visualisation, Figure 4.7 (i.e. My Train Journey) combines the functionality of train-oriented performance statistics with the functionality of journey planning (i.e. My Train Journey could support arbitrary queries about origin-destination pairs, but Fasteroute Delay Explorer could only support direct routes). Moreover, Fasteroute Delay Explorer adopts a colour scale to reflect/indicate the overall performance of a given train over the last few weeks, whereas My Train Journey chooses to directly present a set of selected statistics.

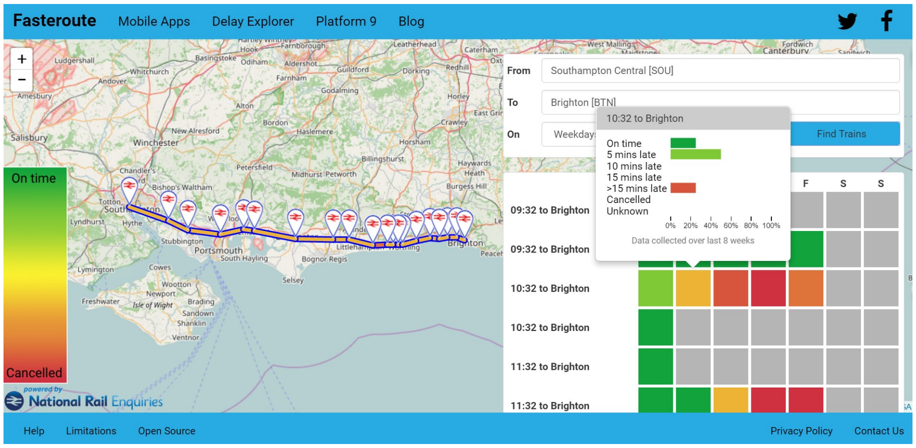


Figure 4.6 Train-oriented performance statistics: Fasteroute (Source: delayexplorer.fasteroute.com/#/, accessed 27 Jan 2017)

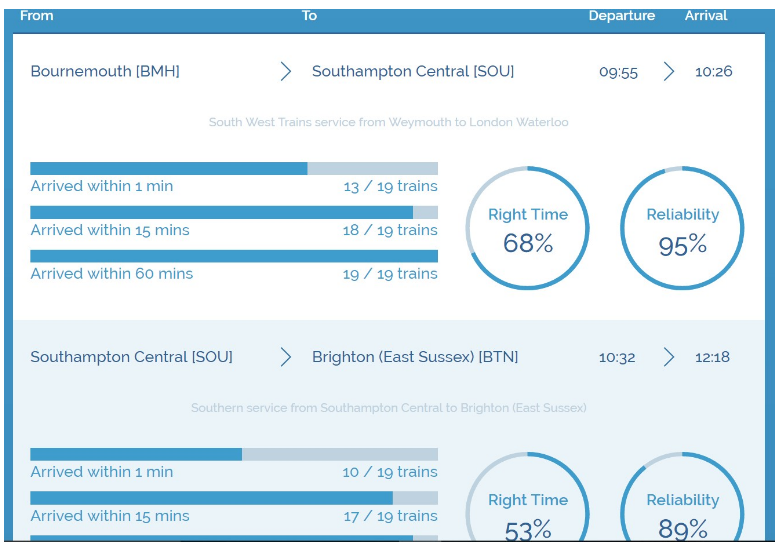


Figure 4.7 Train-oriented performance statistics: My Train Journey (Source: www.mytrainjourney.co.uk/, accessed 27 Jan 2017)

Although these individual-leg-oriented performance statistics could to some degree mitigate the negative effect of missed transfers, they have four potential limitations. Firstly, the information consumers (passengers) have not been truly set free from the burden of computation. Confronted with two or more involved legs, a passenger would still have to estimate the overall performance of a given recommended itinerary by himself/herself, relying heavily on his/her own mathematical ability.

Secondly, even if every user/passenger is good at mathematics, these separately computed statistics tend to hide a lot of key information (e.g. correlation between trains), which impedes passengers' ability to capture the whole picture. Suppose a given recommended itinerary involves two legs with a scheduled transfer time of 5 minutes. And suppose there are 20 past observations (corresponding to 20 observation dates) for each of the two legs to calculate statistics: for the first leg, two of the 20 observations are identified as significant lateness (e.g. > 15 mins late) and the other 18 observations are all found to be on time (< 1 min late); for the second leg, also two are recognised as significant lateness and 18 on time. Then a problem arises: if the two unpunctual observations of the first leg coincide with the two of the second leg (i.e. they happen on the same dates), then the overall punctuality would be 90% (i.e. 18/20); otherwise, the overall punctuality would be 80% (i.e. 16/20).

Thirdly, these individual-leg-oriented performance statistics do not say where a train has lost the time which leads to the delay at the end of its journey. For instance, if the Southampton – Brighton train always departed on time from Southampton and the arrival delays at Brighton were always accumulated en route, a passenger would miss a different number of connections from the scenario in which the arrival delays (at Brighton) were 100% attributable to the departure delays (at Southampton) and no further delay accumulation en route. Lastly, these statistics tend to have limited extensibility and could not provide alternative transfers in the scenario in which the recommended transfer (by a journey planning system) is found to be of poor performance. That is, these separately computed statistics could not provide feedback to a journey planning system to modify the recommended itinerary list when certain of the recommended transfer plans are recognised as unreliable.

4.2.8 The inadequacy of the existing approaches/ideas to tackle Critical Routes

Despite the existence of a variety of algorithmic solutions/ideas to deal with missed transfers (i.e. increasing, to different degrees, the robustness/reliability of the recommendations), none of them could effectively deal with those critical routes (c.f. Sections 3.4 and 3.5) or truly resolve the research problem of this thesis (c.f. Section 1.2).

The potential limitations of the existing solutions/ideas to deal with missed transfers in the general case (c.f. Subsections 4.2.1 – 4.2.7) would also be applicable to the special case of critical routes. Apart from these general limitations/gaps, most of the existing solutions/ideas would either lose their efficacy or result in uncompetitive (/unattractive) recommendations (due to insufficient attention paid to the interplay between competitiveness and reliability) in the special case of critical routes.

Typical examples of losing efficacy when applied to those critical routes include the method of backup information (c.f. Subsection 4.2.3) and the method of robust routing based on historical data (c.f. Subsection 4.2.4). Since these methods implicitly require the existence of multiple (similarly attractive) alternative routes between a given pair of source and target vertices (see Figure 4.8 for an illustration of the method of robust routing based on historical data) but a given critical route would have no such (similarly attractive) alternatives between its two end vertices (c.f. the definition of critical routes in Section 3.5), applying these methods to a given critical route would hardly change the route and itinerary recommendations resulting from current journey planning algorithms (c.f. Sections 3.4 and 3.5) and hence would hardly improve the reliability (/robustness/punctuality) of the recommended itineraries.

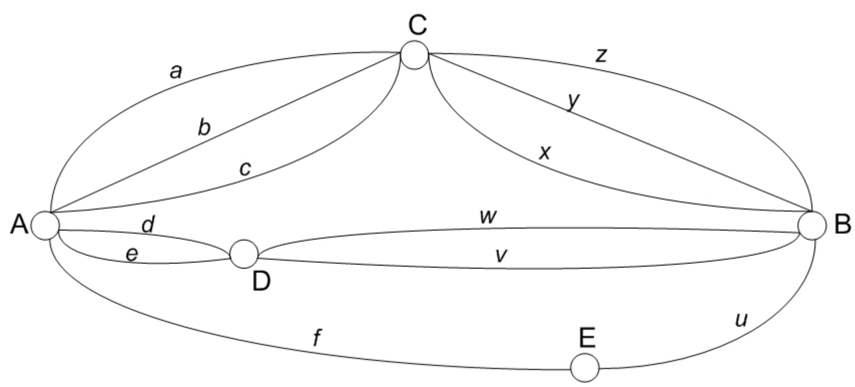


Figure 4.8 An illustration of the core algorithmic idea of Pröger (2016) and Böhmová et al. (2013; 2015)

[Suppose A, B, C, D, and E are five different bus stops within a given urban public transportation network, and *a*, *b*, *c*, *d*, *e*, *f*, *u*, *v*, *w*, *x*, *y*, and *z* are twelve different bus lines. The idea can be decomposed into three major steps. In the first step, all feasible routes (with a constraint of number of transfers) between a given pair of source and target nodes (e.g. A and B) are listed based on the underlying timetable (e.g. <*b*, C, *x*>, <*e*, D, *w*>, and <*f*, E, *u*>). Then, in the second step, the robustness/reliability of each route is assessed and compared (with each other), based on a specified 'latest allowed arrival time' and the analysis of the relevant historical data. In the third step, the route (or several routes) with the best performance in terms of robustness/reliability is (are) selected, and a 'reasonable' departure time (i.e. the latest departure time) for each selected route is calculated from the planned timetable and recommended with its corresponding route.]

Typical examples of resulting in uncompetitive (/unattractive) recommendations when applied to those critical routes include the method of backup information (c.f. Subsection 4.2.3), the method of customisable transfers (c.f. Subsection 4.2.5), and the method of increasing transfer buffers (c.f. Subsection 4.2.6). Since the essence of these methods is either maximising/prioritising the use of those 'safe' transfers (i.e. robust in all (/most) simulated delay scenarios) or making those 'risky' transfers safer (i.e. by adding additional buffers), applying these methods to a given critical route would significantly reduce the competitiveness (/attractiveness) of the recommended itineraries and such a significant reduction in competitiveness (/attractiveness) would render any speculated improvement in reliability (/robustness/punctuality) groundless. Figure 4.9 provides an illustration of the potential effect of applying the method of increasing Minimum Transfer Times (i.e. a light implementation of the method of increasing transfer buffers in Subsection 4.2.6) to a given critical route. Appendix C provides two real-world examples (i.e. two critical routes Knottingley – Wakefield Kirkgate – Nottingham and Ebbw Vale Town – Cardiff Central – Birmingham New Street).

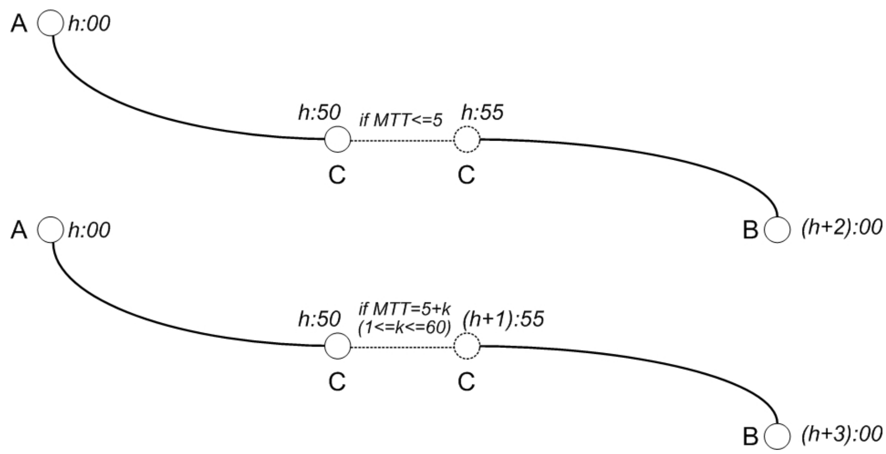


Figure 4.9 An illustration of the potential effect of applying the method of increasing Minimum Transfer Times (MTT) to a given critical route

[Suppose there is a critical route from Station A to Station B via Station C within a given railway network (c.f. the upper curve of the two). The scheduled departure time of the feeder line at A is the start of each hour (denoted by *h:00*) and the feeder leg takes 50 mins in the corresponding schedule. The scheduled departure time of the connecting line at C is 55 past each hour (denoted by *h:55*) and the connecting leg takes 1h05m in the corresponding schedule. That is, the original Minimum Transfer Time (MTT) is no greater than 5 mins for this route. If we increase the original MTT by any value that could result in a new MTT greater than 5 and less than 66 mins, then the new recommendations would become the lower curve of the two. That is, the scheduled journey time would increase from 2h to 3h (a 50% increase), and the change in generalised journey time would be even bigger considering the significant increase in scheduled waiting at C.]

To understand why a significant reduction in competitiveness (/attractiveness) would render any speculated improvement in reliability (/robustness/punctuality) groundless, it is necessary to firstly get an understanding of the interrelationship between competitiveness (/attractiveness) and reliability (/robustness/punctuality).

Ideally, competitiveness and reliability could be simultaneously achieved as long as the scheduled travel time (and other relevant aspects e.g. price and out-of-vehicle waiting) of a given line/route is attractive (compared with the other transport modes e.g. aeroplanes, coaches, private cars, taxis, etc) and the timetable is strictly adhered to (i.e. there exist no disturbances or disruptions). Unfortunately, device malfunctions, human errors, and uncontrollable accidents and weather conditions are omnipresent in the daily operation of trains so that a given timetable can hardly be 100% precisely realised.

The prevalence of delays/variations complicates the interrelationship between competitiveness (/attractiveness) and reliability (/robustness/punctuality). In some/many cases, an improvement in reliability would to some degree contribute to competitiveness, and reliability could be viewed as a component of competitiveness. Such cases correspond to those (highly) repeatable scenarios e.g. commuter routes, direct inter-urban routes, etc. The number of existing passengers who have sufficient (through repeated trials) experiential information about the reliability of those pre-trip itinerary recommendations would be considerable in these scenarios, and hence a reliability improvement would more easily be transmitted to potential passengers by way of word of mouth to influence the (mode, route, or departure time) choices of the potential passengers. In the other cases, the interrelationship between competitiveness (/attractiveness) and reliability (/robustness/punctuality) can be quite different. For example, an improvement in reliability (of those pre-trip itinerary recommendations) in those (highly) repeatable scenarios would have little influence on the (mode, route, or departure time) choices of the existing passengers themselves (e.g. for a given commuter line), for a frequent user could have already had a most realisable itinerary in his/her mind and relies mainly on this self-constructed itinerary (rather than some recommended one) to make choice.

The case of those critical routes (i.e. the research focus of the thesis) is another exception. To facilitate the exposition, the algorithmic solutions/ideas underlying current journey planning systems (i.e. earliest arrival, latest departure, and minimum transfer time, explained in

Sections 3.4 and 3.5) are classified/named as CF (Competitiveness-First) solutions/ideas, while the method of backup information (c.f. Subsection 4.2.3), the method of customisable transfers (c.f. Subsection 4.2.5), and the method of increasing transfer buffers (c.f. Subsection 4.2.6) are classified/named as RF (Reliability-First) solutions/ideas. Based on such a dichotomy, the explanations are as follows. The estimated/predicted reliability improvements by the existing RF solutions/ideas would be achievable if and only if the following assumption (unstated in the relevant literature) could always hold true – a potential passenger who would have otherwise adopted a recommended CF itinerary would also adopt the corresponding (recommended) RF itinerary. This assumption, however, can hardly hold when applying the existing RF solutions/ideas to those critical routes. Let us use Figure 4.9 to facilitate the explanation. If we presume a CF recommendation (c.f. the upper one of the two in Figure 4.9) is competitive (compared with other available modes, routes, and departure times) and can attract some potential passenger onto the track, we can hardly assume/predict with the same confidence that the same person would be attracted onto the track if the corresponding RF itinerary (c.f. the lower one of the two in the figure) has been recommended (instead of the CF one). Why? Because the RF version has a significantly longer estimated (/scheduled/advertised) travel time than the corresponding CF version – 50% longer w.r.t. scheduled (/advertised) journey time (i.e. (3h-2h)/2h) and more than 50% longer w.r.t. generalised journey time (the extra 1h adds to out-of-vehicle waiting). Why would the speculated reliability improvements by the existing RF solutions/ideas be groundless? Two reasons. On the one hand, if we assume there exists a customer base (believed to be much smaller than that for e.g. a commuter line) for a critical route, then a speculated improvement in reliability (by adopting the existing RF solutions/ideas) would have little influence on the (mode, route, or departure time) choices of either the customer base (i.e. frequent users) or the potential passengers – a frequent user could have had (through repeated trials) and adopted his/her own self-constructed itinerary (rather than some recommended one) and a potential passenger could hardly obtain the experiential information about reliability from a frequent user due to the much lower exposure to such information (than in the case of e.g. commuter routes, direct inter-urban routes, etc). On the other hand, if we assume there exists no customer base but instead exists a group of potential passengers for a critical route, then the speculated reliability improvements (by adopting the existing RF solutions/ideas) would also be a rubber cheque – if a potential customer could not be firstly attracted onto the track, he/she would never have the experiential information about the reliability of those pre-trip itinerary recommendations and the speculated reliability improvements would become meaningless. Note that the above analysis of the interrelationship between competitiveness (/attractiveness) and reliability (/robustness/punctuality) in the context of critical routes (i.e. competitiveness should be given a higher priority than reliability in such scenarios) does not mean competitiveness (/attractiveness) should be maximally pursued without allowing for the potential impact on reliability (/robustness/punctuality) – a highly competitive itinerary recommendation (e.g. generated by the existing CF solutions/ideas) may attract many potential passengers onto the track but its poor reliability would impede the next cooperation between these passengers and the relevant railway companies on the same route or even on the other transfer-involved routes.

4.3 A historical-data-based approach tailored for tackling Critical Routes

4.3.1 Central idea: less is more

By reviewing the existing information-based approaches to tackling transfer-related problems, we can get a glimpse of the design philosophy adopted by these approaches: almost all of them are based on 'presumption of guilt'. That is, every possible transfer plan within a timetable-based transport network is treated (by these approaches) as potentially unreliable due to the impact of train delays and disruptions, and hence additional information should be provided about performance statistics or alternative plans for every transfer plan to enable passengers to make better choices. In realistic applications, however, this 'holistic treatment' of transfer plans not only increases the computational complexity of the underlying algorithms but also is likely to result in uneasiness or overreactions of information consumers (passengers). Indeed, no transfer plan can be said to be 100% reliable due to the fact that there are lots of endogenous and exogenous factors influencing train movements. In reality, however, most transfer plans recommended by a journey planning system can be realised with a considerably high degree of certainty due to the mechanism of minimum transfer times (c.f. Subsection 3.4.2). Therefore, it seems unnecessary or even misleading to provide additional information (warning) about those considerably reliable transfer plans. After all, low probabilities tend to be overweighted when losses are expected (Kahneman, 2012), which may cause inconvenience to information consumers (passengers).

Based upon the above considerations, a historical-data-based approach is proposed (see Figure 4.10 for an overview), the design philosophy of which is embodied by the following four aspects:

- Less consumption of computing resources and less disturbing information: it can be seen from Figure 4.10 that the biggest difference between the existing approaches and the approach proposed here lies in that the 'local/precision treatment' of the identified critical routes is adopted here, rather than the 'holistic treatment' adopted in the existing approaches. This difference is originated from the difference between design philosophies: the philosophy of 'less is more' is adopted here, based on the consideration that every day people are confronted with too many pieces of irrelevant and unnecessary information in such an age of information explosion, and providing additional information about those low-risk transfer plans would be disturbing. From the perspective of energy consumption, an algorithm requiring less computing resources would be more eco-friendly than those computationally intensive ones.

- Less reliance on past experience: since the proposed approach is historical-data-based, it does not presume that passengers have sufficient experience of train delays or disruptions (compared with the approach of 'customisable transfers' in Subsection 4.2.5). In fact, those passengers choosing a critical route tend to be less likely to have sufficient experience of the performance of such a route due to the fact that a critical route is long-distance and transfer-involved. Therefore, a presumption of inexperienced passengers would be more appropriate.

- Less requirement on mathematical ability: as is to be introduced later in Algorithm 3 and Algorithm 4, the proposed approach is based on performance statistics that are oriented toward a whole journey (itinerary) rather than toward individual service legs (compared with the approach mentioned in Subsection 4.2.7). Therefore, it would be able to set the information consumers (passengers) free from those demanding calculations by themselves.

- Less inconvenience for passengers to integrate fragmented information by themselves: as is to be illustrated later in this chapter, the proposed approach has great potential to be integrated into the existing journey planning algorithms (and hence the corresponding journey planning systems), which would be able to enhance the functionality of the existing systems and facilitate the dissemination of this additional information about itinerary-level performance. By comparison, the approach of individual-leg-oriented performance statistics (c.f. Subsection 4.2.7) has less extensibility. That is, those individual-leg-oriented performance statistics are limited to providing descriptive information: they could not be utilised by the existing journey planning algorithms to provide prescriptive information about alternative plans or predicted arrival/journey times due to a lack of itinerary-level performance indicators.

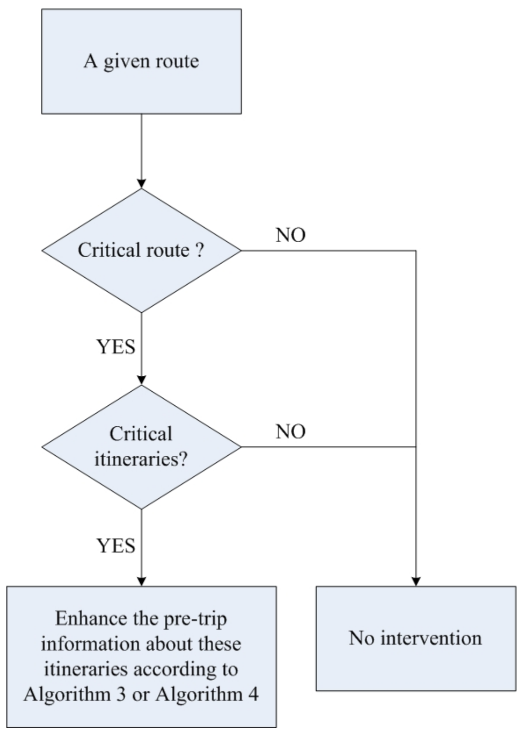


Figure 4.10 An overview of the proposed algorithmic approach

4.3.2 IPS vs. PBPM: two sides of one coin

The previous subsection briefly describes the central idea of the proposed algorithmic approach: 1) ‘local treatment’ of the identified critical routes; and 2) itinerary-oriented performance evaluation. In the end of the previous subsection (c.f. Figure 4.10), two algorithms are mentioned: Algorithm 3 and Algorithm 4, which comprise the core part of the proposed algorithmic approach. This subsection is to present the pseudo code of Algorithm 3 (IPS) and Algorithm 4 (PBPM). Both IPS and PBPM are historical-data-based and involve quite similar data pre-processing steps. However, there is a major difference between the two algorithms in terms of the specific statistics adopted and the representation of results. Roughly speaking, IPS can be viewed as an augmented version of those individual-leg-oriented performance statistics (c.f. Subsection 4.2.7 for details about those individual-leg-oriented performance statistics), whereas PBPM is inspired by the representation of real-time delay alerts in the existing real-time information systems (to be detailed later in Subsection 4.3.8).

Algorithm 3 below presents the pseudo code of Itinerary-oriented Performance Statistics (IPS). As its name implies, IPS is designed to calculate and present itinerary-oriented performance statistics for each critical itinerary following a given critical route (on a given query date). In order to obtain such itinerary-oriented performance statistics, detailed historical train movements data should be available and several involved parameters should be pre-determined (e.g. NTT, AW, etc). Moreover, data pre-processing and visualisation are also important.

Algorithm 3: IPS (Itinerary-oriented Performance Statistics)

Input: a sufficiently large sample of detailed historical train movements data about a given critical route (in recent past)

Output: a recommended itinerary list (for a particular date in the near future) in which each critical itinerary is associated with an itinerary-specific performance statistic

// Step 1: construct a route-view timetable (**RVT**) for the studied route
identify all the involved service legs along the critical route
extract from historical train movements data all the relevant information about each service leg (train identifier, run date, station identifier, scheduled arrival/departure times, recorded arrival/departure times, platform, cancellation, etc)
merge the data records of the involved service legs into a **RVT** by concatenating the corresponding services running on the same dates and following the scheduled order
sort **RVT** by date and scheduled departure time

// Step 2: calculate the net transfer time (NTT) and the actual window (AW) of each
// involved transfer
for each record i (corresponding to a critical itinerary) in **RVT**:
    for each involved transfer j:
        add into **RVT** two new columns NTT_j and AW_j
        $\mathrm{NTT}_{i,j} = \mathrm{DIST}_{i,j} / \mathrm{SPEED}_{i,j}$

```
        store NTT_{i,j} into NTT_j
        if cancelled(T_j) == True or cancelled(T_{j+1}) == True:
            AW_{i,j} = -∞
        else:
            AW_{i,j} = dep_a(T_{j+1}) – arr_a(T_j)
        store AW_{i,j} into AW_j

// Step 3: calculate itinerary-oriented performance statistics based on some predefined
//          threshold value
group the records in RVT by scheduled departure time
for each group g in RVT:
    for each record i:
        flag = 1
        for each involved transfer j:
            if AW_{i,j} < NTT_{i,j} :
                flag = 0
                break
        if arr_a(T_k) – arr_s(T_k) ≥ TAL:
            flag = 0
        if flag = 1:
            success(g) + 1
        otherwise:
            failure(g) + 1
    IPS_g = success(g) / (success(g) + failure(g))
    store IPS_g into IPS

// Step 4: construct and display the enhanced itinerary list
construct the recommended itinerary list RIL_DATE for a given query date DATE (in
the near future) based on timetable data
calculate IPS from the latest historical data according to Steps 1 – 3
associate each critical itinerary in RIL_DATE with its corresponding value in IPS
return RIL_DATE
terminate
```

Generally, the above algorithm (i.e. IPS) can be subdivided into four steps (see the pseudo code in Algorithm 3): Steps 1 – 3 belong to back-end development (i.e. data processing), while Step 4 belongs to front-end development (i.e. user interface design). A Python implementation of the back end (core part) of IPS (and also PBPM in Algorithm 4) is presented in Appendix D, the source code of which comprises approximately 600 lines of commands. It should be noted that although front-end development is very important, this section is mainly focused on the back end.

Algorithm 4 below presents the pseudo code of Performance-Based Pre-Modification of advertised arrival times (PBPM). Unlike IPS, PBPM abandons the representation of pure statistics (i.e. probabilities, c.f. Figures 4.6 and 4.7 in Subsection 4.2.7) and adopts a method of modifying well in advance the advertised arrival time of each critical itinerary following a given critical route (on a given query date). Roughly speaking, PBPM adds to each critical itinerary extra allowance (i.e. time supplement) to reduce the impact of delays/disruptions, based on the historical performance of each particular itinerary. In order to implement PBPM, detailed historical train movements data should be available and several involved parameters should be pre-determined (e.g. NTT, AW, etc). Moreover, data pre-processing and several heuristics are also involved.

In the pseudo code of IPS and PBPM, all the notations in italics are one-dimensional list (array) objects, those in bold are two-dimensional tables, and uppercase letters are constant parameters. Each step in the pseudo code is to be explained later in the subsequent subsections. The relevant symbols are as follows:

- $DIST_{i,j}$ and $SPEED_{i,j}$ respectively represent the (horizontal and vertical) distance between a pair of feeder and connecting trains <i, j> and the walking speed of an average passenger between the feeder train i and the connecting train j.
- cancelled(·) is an indicator variable to judge whether a given train was cancelled.
- $dep_s(\cdot)$, $arr_s(\cdot)$, $dep_a(\cdot)$, $arr_a(\cdot)$, and $arr_m(\cdot)$ respectively represent the scheduled departure time of, the scheduled arrival time of, the actual departure time of, the actual arrival time of, and the modified arrival time of a given train/itinerary.
- flag is an indicator variable to control the execution of the relevant for-loops.
- TAL is short for Threshold for Arrival Lateness.
- success(·) and failure(·) are counter variables that respectively represent the number of successful and unsuccessful realisations of a given itinerary.
- $p_0(\cdot)$ represents the success rate of a given itinerary.
- $\delta(\cdot)$ represents the average delay of a given itinerary at the target/destination station.
- $jt_0(\cdot)$ and $jt_1(\cdot)$ respectively represent the average journey time of a given itinerary in the scenario in which no missed transfers and the average journey time of a given itinerary in the scenario in which there is exactly one missed transfer.
- $\mathbf{RIL_{DATE}}$ means the Recommended Itinerary List for a given query date.
- $HEADWAY_{avg}$ means the average headway of the involved lines in a given itinerary.

Algorithm 4: PBPM (Performance-Based Pre-Modification of advertised arrival times)

Input: a sufficiently large sample of detailed historical train movements data about a given critical route (in recent past)

Output: a recommended itinerary list (for a particular date in the near future) in which the advertised arrival time of each critical itinerary is modified well in advance based on itinerary-specific performance in history

```
// Step 1: construct a route-view timetable (RVT) for the studied route
identify all the involved service legs along the critical route
extract from historical train movements data all the relevant information about each
service leg (train identifier, run date, station identifier, scheduled arrival/departure
times, recorded arrival/departure times, platform, cancellation, etc)
merge the data records of the involved service legs into a RVT by concatenating the
corresponding services running on the same dates and following the scheduled order
sort RVT by date and scheduled departure time

// Step 2: calculate the net transfer time (NTT) and the actual window (AW) of each
//         involved transfer
for each record i (corresponding to a critical itinerary) in RVT:
    for each involved transfer j:
        add into RVT two new columns NTT_j and AW_j
        NTT_{i,j} = DIST_{i,j} / SPEED_{i,j}
        store NTT_{i,j} into NTT_j
        if cancelled(T_j) == True or cancelled(T_{j+1}) == True:
            AW_{i,j} = -∞
        else:
            AW_{i,j} = dep_a(T_{j+1}) – arr_a(T_j)
        store AW_{i,j} into AW_j

// Step 3: calculate the probability of missed transfers for each critical itinerary
group the records in RVT by scheduled departure time
for each group g in RVT:
    for each record i:
        flag = 1
        for each involved transfer j:
            if AW_{i,j} < NTT_{i,j} :
                flag = 0
                break
        if flag = 1:
           success(g) + 1
        otherwise:
           failure(g) + 1
    p_0(g) = success(g) / (success(g) + failure(g))
    store p_0(g)  into p_0

// Step 4: calculate the average lateness at the destination station for the k^th involved
//         leg of each critical itinerary
group the records in RVT by scheduled departure time
for each group g in RVT:
    for each record i:
```

```
        Δ(T_k) = arr_a(T_k) – arr_s(T_k)
        store Δ(T_k) into Δ(g)
    δ(g) = Δ(g).average()
    store δ(g) into δ

// Step 5: modify the advertised arrival time of each critical itinerary and display
construct the recommended itinerary list RIL_DATE for a given query date DATE (in
the near future) based on timetable data
calculate p0 and δ from the latest historical data according to Steps 1 – 4
for each critical itinerary i_c in RIL_DATE:
    look up the corresponding p_0(i_c) and δ(i_c) in p_0 and δ
    jt_0(i_c) = arr_s(T_k) – dep_s(T_1) + δ(i_c)
    jt_1(i_c) = arr_s(T_k) – dep_s(T_1) + HEADWAY_avg + δ(i_c)
    arr_m(i_c) = dep_s(T_1) + p_0(i_c)·jt_0(i_c) + (1 – p_0(i_c))· jt_1(i_c)
return RIL_DATE
terminate
```

Note that IPS (Algorithm 3) has been created and presented mainly as an introductory algorithm to the proposed solution (i.e. PBPM/Algorithm 4) of the core research problem. The inclusion of this particular approach into the thesis has been mainly due to the consideration that it might help better understand the underlying statistical ideas and technicalities of PBPM.

PBPM (Algorithm 4) has been created and adopted to enhance the pre-trip information about those critical (transfer-involved) itineraries (corresponding to some identified critical route), which has simultaneously taken into account the constraint of capacity utilisation and the interplay between the competitiveness (/attractiveness) of and the reliability (/punctuality/robustness) of the recommended itineraries (to be further explained in Subsection 5.3.12).

4.3.3 Sample size

The input of IPS and PBPM is a 'sufficiently large' sample of detailed historical train movements data about a given critical route. Here, a question arises: how large is 'sufficiently large'? This is a big topic and is an unavoidable question for any statistical method. According to the law of large numbers (c.f. Section 2.4), the sample size should be as large as possible. However, there is not a one-size-fits-all answer within the field of probability and statistics. And the sample size adopted in realistic applications is often restricted by the availability of the relevant (historical) data. Here, a 'sufficiently large' sample can be thought of as a collection of several-months historical data, based on the consideration about the availability of the relevant data and the treatment adopted in the existing real-world applications (c.f. Subsection 4.2.7). Further discussion about the appropriate sample size is to be presented later in the next chapter.

4.3.4 Route-View Timetable (RVT)

Step 1 in Algorithm 3 and Algorithm 4 is mainly for data pre-processing. In order to help understand this process, Figures 4.11 and 4.12 below are employed to provide an illustration. Both of the two illustrative examples are based on historical train movements data collected from Realtime Trains (RTT, a real-time passenger information system in Britain, c.f. Subsection 3.7.2). While Figure 4.11 offers a birds-eye view of the data pre-processing in Step 1, Figure 4.12 provides a more concrete example of how to obtain a Route-View Timetable (RVT).

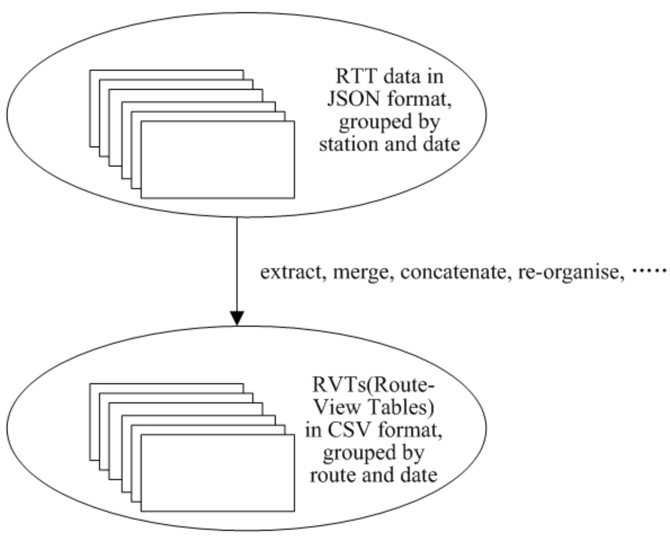


Figure 4.11 A general illustration of Step 1 in Algorithm 3

The raw data (at the top of Figure 4.11) is a set of collected RTT data. These RTT data are grouped by station and date. Each file in this set contains information about all the arrival and/or departure events that have happened at a given station on a given day (the top of Figure 4.12 gives such an illustration). The exact number of files in the RTT data set depends on how many stations and how many days are involved in the studied route. Figure 4.12 uses the route Bournemouth-Southampton-Brighton (BSB) as an example and the study period is from 12 Oct 2015 to 04 Dec 2015 (8 weeks). If this context is adopted in Figure 4.11, then the RTT data set would contain 120 JSON files (40 working days in total and 3 stations involved each day).

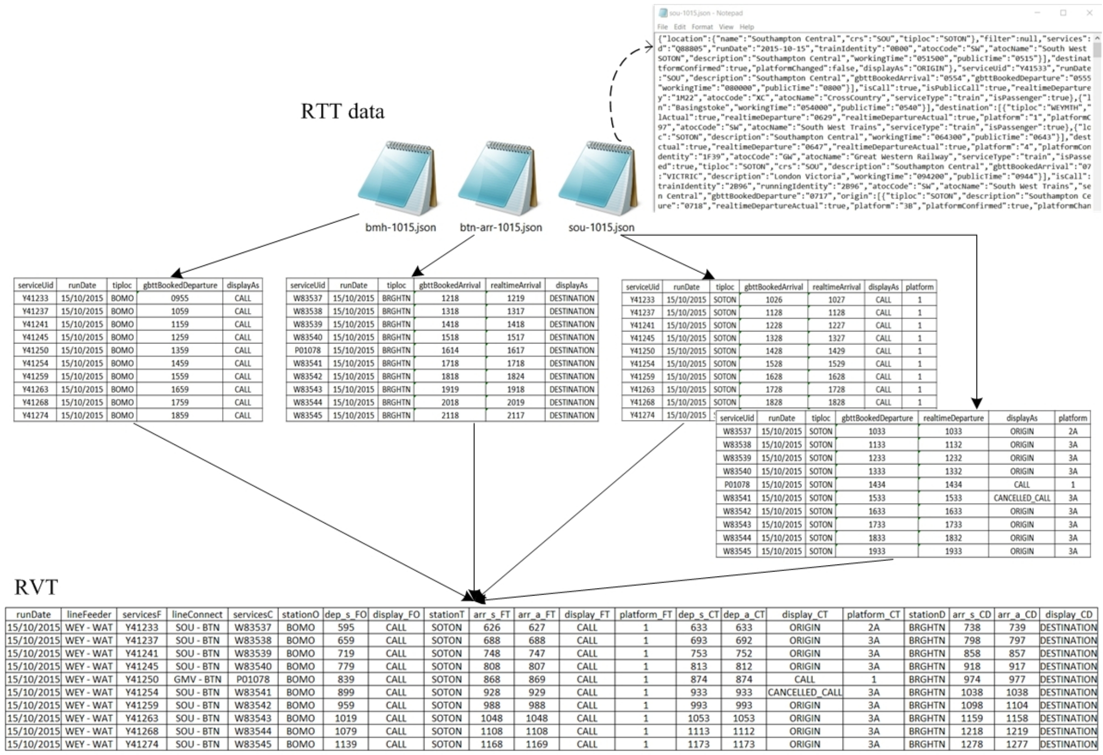


Figure 4.12 A more concrete illustration of Step 1 in Algorithm 3

Through a series of data processing sub-steps, the 120 JSON files in the RTT data set are converted into 40 Route-View Timetables (RVTs) (see Figure 4.11 and Figure 4.12). These RVTs are grouped by route and date (here only one route is considered, so the RVT set is only grouped by date). Each RVT contains all the necessary information about all of the studied (critical) itineraries (see Figure 4.12 for example).

There is an illustrative RVT in Figure 4.12, and Table 4.1 below is a magnified version of that in Figure 4.12. In this table, each row corresponds to a studied (critical) itinerary and each column corresponds to the value of a specific attribute. All of these ten studied itineraries belong to the route Bournemouth-Southampton-Brighton, and they all happened on 15 Oct 2015.

Table 4.1 An example of RVT (Route-View Timetable)

runDate	lineFeeder	servicesF	lineConnect	servicesC	stationO	dep_s_FO	display_FO	stationT	arr_s_FT	arr_a_FT	display_FT	platform_FT	dep_s_CT	dep_a_CT	display_CT	platform_CT	stationD	arr_s_CD	arr_a_CD	display_CD
15/10/2015	WEY - WAT	Y41233	SOU - BTN	W83537	BOMO	595	CALL	SOTON	626	627	CALL	1	633	633	ORIGIN	2A	BRGHTN	738	739	DESTINATION
15/10/2015	WEY - WAT	Y41237	SOU - BTN	W83538	BOMO	659	CALL	SOTON	688	688	CALL	1	693	692	ORIGIN	3A	BRGHTN	798	797	DESTINATION
15/10/2015	WEY - WAT	Y41241	SOU - BTN	W83539	BOMO	719	CALL	SOTON	748	747	CALL	1	753	752	ORIGIN	3A	BRGHTN	858	857	DESTINATION
15/10/2015	WEY - WAT	Y41245	SOU - BTN	W83540	BOMO	779	CALL	SOTON	808	807	CALL	1	813	812	ORIGIN	3A	BRGHTN	918	917	DESTINATION
15/10/2015	WEY - WAT	Y41250	GMV - BTN	P01078	BOMO	839	CALL	SOTON	868	869	CALL	1	874	874	CALL	1	BRGHTN	974	977	DESTINATION
15/10/2015	WEY - WAT	Y41254	SOU - BTN	W83541	BOMO	899	CALL	SOTON	928	929	CALL	1	933	933	CANCELLED_CALL	3A	BRGHTN	1038	1038	DESTINATION
15/10/2015	WEY - WAT	Y41259	SOU - BTN	W83542	BOMO	959	CALL	SOTON	988	988	CALL	1	993	993	ORIGIN	3A	BRGHTN	1098	1104	DESTINATION
15/10/2015	WEY - WAT	Y41263	SOU - BTN	W83543	BOMO	1019	CALL	SOTON	1048	1048	CALL	1	1053	1053	ORIGIN	3A	BRGHTN	1159	1158	DESTINATION
15/10/2015	WEY - WAT	Y41268	SOU - BTN	W83544	BOMO	1079	CALL	SOTON	1108	1108	CALL	1	1113	1112	ORIGIN	3A	BRGHTN	1218	1219	DESTINATION
15/10/2015	WEY - WAT	Y41274	SOU - BTN	W83545	BOMO	1139	CALL	SOTON	1168	1169	CALL	1	1173	1173	ORIGIN	3A	BRGHTN	1278	1277	DESTINATION

The two columns ‘lineFeeder’ and ‘lineConnect’ in Table 4.1 tell us which two train lines are involved to complete a given itinerary (only one transfer is involved in this example; an RVT could also be constructed in a similar way to cover more complex transfer scenarios). In this example, all of the ten itineraries have the same feeder line: the Weymouth-London Waterloo line operated by South West Trains. Apart from the one having a value of ‘P01078’ under the ‘serviceC’ column, all of the other nine itineraries have the same connecting line: the Southampton-Brighton line run by Southern. The only exception involves the Great Malvern-Brighton line operated by Great Western Railway.

The two columns ‘serviceF’ and ‘serviceC’ tell us the service id numbers of the two involved trains for a given itinerary. A service id number is generally stable and unique across the whole network during a given timetable period. For example, the string in the first cell under ‘serviceF’ (i.e. ‘Y41233’) represents the South West Trains service that runs from Weymouth at 09:03 each weekday morning to London Waterloo during the period 05 Oct 2015 – 11 Dec 2015.

The three columns ‘stationO’, ‘stationT’, and ‘stationD’ store the names of the origin station (i.e. BOMO in this example), the transfer station (i.e. SOTON), and the destination station (i.e. BRGHTN), respectively.

The four columns ‘dep_s_FO’, ‘arr_s_FT’, ‘dep_s_CT’, and ‘arr_s_CD’ store the values of the scheduled (nominal) departure time of the feeder train at the origin station, the scheduled arrival time of the feeder train at the transfer station, the scheduled departure time of the connecting train at the transfer station, and the scheduled arrival time of the connecting train at the destination station, respectively. Note that all the (scheduled and actual) stop times have been converted into integers (bounded by [0, 1439]) to enable the calculation of travel times, waiting times, etc. The conversion is based on the following simple algorithm: each integer in the interval [0, 1439] corresponds to the difference in minutes between the very beginning of the day (i.e. 00:00) and the given stop time (e.g. 595 ↔ 09:55).

The three columns 'arr_a_FT', 'dep_a_CT', and 'arr_a_CD' contain the values of the recorded (actual) arrival time of the feeder train at the transfer station, the recorded departure time of the connecting train at the transfer station, and the recorded arrival time of the connecting train at the destination station, respectively.

The four columns 'display_FO', 'display_FT', 'display_CT', and 'display_CD' contain information about the status of a given train at a given station. Valid values in these columns include CALL, ORIGIN, DESTINATION, STARTS, TERMINATES, and CANCELLED_CALL. ORIGIN and DESTINATION represent the original origin and destination of a service, respectively. If STARTS or TERMINATES appear, then this means a service has started short or terminated en-route, and meanwhile the original origin/destination will show CANCELLED_CALL. This status information is useful in the calculation of performance statistics in subsequent steps of Algorithm 3 and Algorithm 4.

The remaining two columns 'platform_FT' and 'platform_CT' respectively store the information about the allocated platform for the feeder train at the transfer station and the allocated platform for the connecting train at the transfer station. This piece of information about platform allocation is potentially useful in dealing with the impact of platform changes on the estimation of NTT (Net Transfer time) in subsequent steps.

The above example is only for one day. It needs to be combined with other daily RVTs to form an N-week sample ($N = 4, 6, 8, 12$, etc). A sample table for this studied route (i.e. BSB) normally contains $50N$ records (10 per day and $5N$ working days), but in rare cases the number of records would be slightly smaller than $50N$ (e.g. $50N - 1$, $50N - 2$, etc) due to the cancellation of some service(s) on a particular day (i.e. the train movements data are completely missing for the service(s)). Further discussion about train cancellations can be found later in Subsection 4.3.6.

4.3.5 Net transfer time (NTT)

Steps 2 and 3 are responsible for the majority of computations of Algorithm 3, and are an indispensable component of Algorithm 4. Three key parameters are involved in these two steps: net transfer time (NTT) and actual window (AW) in both Algorithm 3 and Algorithm 4, and threshold for arrival lateness (TAL) in Algorithm 3. In the following, the considerations are presented about how to determine these parameters based on available information.

Determining the NTT (c.f. Subsection 3.5.2) for each critical transfer that is involved in each critical itinerary (following a given critical route) can be a heavy task if taking into account the various factors potentially influencing passengers' transfer activities (e.g. platform changes, level of crowdedness in the station, boarding/alighting locations, etc). In reality, however, the determination of NTT is not that difficult due to the following four reasons.

Firstly, according to the definition of NTT (c.f. Subsection 3.5.2, 'physically possible minimum time required to walk from T1 to T2 within the station'), a 'free-flow' walking speed and the shortest walking path can be adopted without the need for considering in-station congestions. Since a calculated NTT has a precision of one second, it can then be converted into minutes (to conform to the granularity of a timetable) by rounding it up to the nearest integer, which is equivalent to add to itself allowances to enable an average passenger to successfully complete the transfer.

Secondly, due to the periodicity of train schedules (c.f. Subsection 2.5.2), the transfer(s) involved in each critical itinerary often follows the same pattern. That is, the platform allocation often remains the same between different hours of a day (c.f. the two columns 'platform_FT' and 'platform_CT' in Table 4.1), and hence it is often enough to determine a route-specific NTT rather than to determine a set of connection-specific NTTs.

Thirdly, the influence of platform changes on the determination of NTTs is also found to be limited, based on extensive analysis of historical train movements data from British railways: a close examination of the large sample (about 1.4 million valid observations) of 12-months train movements data (c.f. Section 3.7) reveals that the probability of a platform change (i.e. an incoming train is rerouted within a station) is approximately 5% within Britain's passenger rail system (which can be regarded as low-probability events). In practical applications, the following strategy can be adopted to reduce the impact of platform changes: if a given sample is found to contain many platform changes in data pre-processing, then scenario-specific NTTs can be assigned to each scenario; otherwise, a route-specific NTT is enough.

Lastly, a reference point can be chosen (for each of the two involved platforms) to simplify the estimation of the walking distance (e.g. choosing the midpoint of each platform as the reference point).

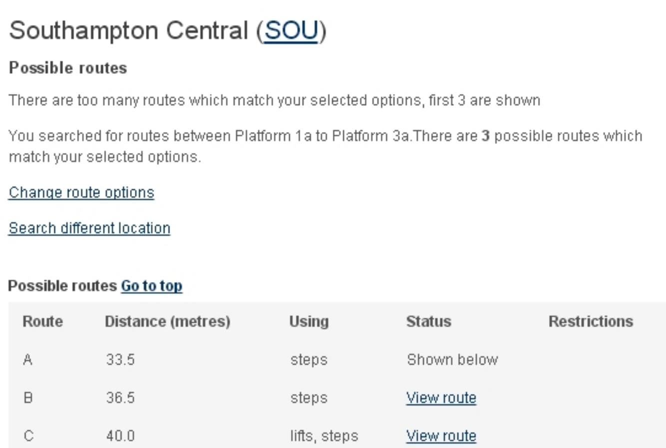
Based on the above considerations, the determination of an NTT (for a given transfer) is reduced to the determination of two parameters: the distance between two platforms and the walking speed within the station (c.f. Step 2 in Algorithm 3 and Algorithm 4). Below is an illustrative example of how to determine the walking distance and walking speed in the context of the route Bournemouth – Southampton – Brighton.

As mentioned previously, in practice it is enough to determine a route-specific NTT when the studied critical transfers follow the same transfer pattern. That is, these transfers happen at the same station (i.e. Southampton Central in this context) and each of the two involved legs stops at the same platform between different hours of a day. From Table 4.1 we can see that most of the studied transfers follow the pattern Platform 1 to Platform 3A (the two exceptions are to be dealt with later). Therefore, we can either carry out fieldwork to determine the NTT between Platform 1 and Platform 3A, or simply exploit the station layout information from the Internet to estimate this parameter.

Figures 4.13 and 4.14 below give an example of using NRE to determine this parameter. For the example considered here, we can simply use the 'plan a route' functionality on the station information page of Southampton Central (Figure 4.13) to enquire about the distance between the two involved platforms to estimate the route-specific NTT. As shown in Figure 4.13, if we choose platform 1a as the origin and platform 3a as the destination, we can get a list of recommended routes within the station. And if we choose the shortest one (route A in the figure), then the distance between the two platforms can be estimated by the planar distance between the two platforms (33.5 metres in this case), plus the steps involved (54 steps in this case, see Figure 4.14). Here, if a walking speed of 5 km/h (about 1.39 m/s) is adopted on flat ground and 2 steps per second (this approximation of walking on stairs is based on fieldwork by the author and the empirical results presented in literature such as Fujiyama and Tyler,

2010) is adopted for climbing up and down, then we can obtain an estimated total walking time of 51 seconds. And if considering the extra time spent on looking for the information about the relevant platforms during the transfer process, we can add another 5 seconds to the NTT, which results in an estimation of 56 seconds between Platform 1 and Platform 3A. For the two exceptions (i.e. 1 → 2A and 1 → 1), their NTTs are estimated to be no more than the NTT between Platform 1 and Platform 3A (see Figure 4.14) but are also impossible to be 0 (a transfer will always consume a certain amount of time, no matter how little the exact amount is). Therefore, a rough estimation of the NTTs for the two exceptional situations can be obtained, which lies between 0 and 56s. Based on the above estimations and the fact that historical train movements data (i.e. RTT data in this context) often have a precision of one minute (i.e. the granularity of these historical data is one minute), a unified NTT of 1 minute can be assigned to this route.

It should be noted that the ‘plan a route’ functionality on NRE’s station information pages has been removed at the time of writing this thesis. As an alternative way to estimate the planar distance between two platforms within a transfer station, the ‘measure distance’ functionality of Google Maps (Figure 4.15) can also be utilised to estimate NTTs, with the aid of NRE’s station information about number of steps (Figure 4.14). And if these online resources cannot meet the need for precision, fieldwork can be conducted (in fact, previous research has shown that small estimation errors in NTTs have a limited impact on the obtained results of itinerary-oriented performance statistics; c.f. Guo and Preston, 2016).


Southampton Central (SOU)

Possible routes

There are too many routes which match your selected options, first 3 are shown

You searched for routes between Platform 1a to Platform 3a.There are **3** possible routes which match your selected options.

Change route options

Search different location

Possible routes Go to top

Route	Distance (metres)	Using	Status	Restrictions
A	33.5	steps	Shown below	
B	36.5	steps	View route	
C	40.0	lifts, steps	View route	

Figure 4.13 NRE’s station information page: Example One (Source: www.nationalrail.co.uk/ , accessed: 25 Oct 2015)

Figure 4.14 NRE's station information page: Example Two (Source: www.nationalrail.co.uk/ , accessed: 25 Jan 2017)

Figure 4.15 Using Google Maps to estimate walking distance within a railway station (Source: www.google.co.uk/maps/ , accessed 25 Jan 2017)

4.3.6 Actual window (AW) and threshold for arrival lateness (TAL)

Apart from NTT, AW (actual window) is another important parameter in the determination of whether a scheduled transfer is missed (by most of the relevant passengers) on a particular date in the past (c.f. Step 2 in Algorithm 3 and Algorithm 4). Compared with the determination of NTTs, the determination of AWs is relatively straightforward: they can be calculated directly from the recorded (actual) arrival/departure times in a route-view timetable (RVT). For example, if a feeder train is recorded to have arrived at the transfer station at 11:05 a.m. on a particular day (in the past) and the corresponding connecting train is recorded to have departed from the same station at 11:07 a.m. on the same day, then the AW of this pair of feeder and connecting trains is 2 minutes. And suppose the NTT for this transfer is 1 minute, and then this particular transfer is regarded as not missed.

Although the determination of AWs is generally straightforward, special attention needs to be paid to those scenarios in which train cancellations have been recorded. As previously mentioned in the explanation of Route-View Timetable (RVT), train status information can also be retrieved from the historical train movements data (i.e. RTT data in this context, c.f. Subsection 4.3.4). That is, it can be learned from the available historical data about whether a scheduled train arrival/departure event at a particular station is cancelled. Therefore, if a feeder/connecting train is recorded as 'CANCELLED_CALL' on a particular date in the past (i.e. the scheduled arrival/departure event has been cancelled on that day), then the corresponding transfer is regarded as invalid/missed. At the algorithmic level, these cancellation-involved scenarios can be dealt with by assigning to them some special value (i.e. minus infinity in Step2 of Algorithms 3 and 4) to distinguish them from the others.

Apart from NTT and AW, the threshold for arrival lateness (TAL) is another important parameter in the calculation of itinerary-oriented performance statistics, which is involved in Step 3 of Algorithm 3 (c.f. Subsection 4.3.2). The reason why it is necessary to introduce TAL into Algorithm 3 lies in that if such a threshold is not predefined and the algorithm only checks whether each involved transfer has been successfully realised, then those calculated itinerary-oriented performance statistics would be biased. Figure 4.16 below provides an illustration of the importance of introducing TAL into the calculation of itinerary-oriented performance statistics.

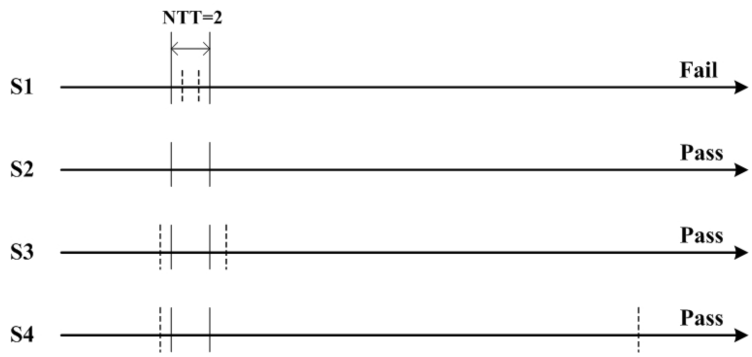


Figure 4.16 An illustration of why it is necessary to predefine a threshold for arrival lateness

In the four scenarios (i.e. S1 – S4) presented in Figure 4.16, the horizontal lines represent the time axis, the two solid vertical lines (in each of the four scenarios) represent the net transfer time for this studied transfer (NTT = 2 minutes), and the two dotted vertical lines (they are masked by the solid vertical lines in S2) represent the actual window (AW) between this pair of feeder and connecting trains. If a threshold for arrival lateness is not predefined, then the realised connection in S4 would pass the test for a 'successful realisation' of the itinerary (assuming that this itinerary involves only one transfer). But in fact this realisation is based on the significant lateness of the connecting leg and hence should not be counted as a 'successful realisation'.

But how to determine an appropriate TAL? Here the consideration is that some industry standards can be adopted as the threshold. Recall that Subsection 2.5.4 has introduced the operational practice in European railways: heuristic measurements are widely adopted by European railways to conduct network-level performance evaluation in terms of punctuality and reliability, the mechanism of which is similar to TAL. Also, those individual-leg-oriented performance statistics in Figures 4.6 and 4.7 (c.f. Subsection 4.2.7) are based on predefined TALs. In fact, it is almost impossible to carry out quantitative analysis or performance evaluation/comparison without some preset standard.

After an investigation of the relevant definitions of Network Rail (c.f. Network Rail, 2017), four candidate thresholds are identified: 1 minute, 5 minutes, 10 minutes, and 30 minutes. According to Network Rail's statistical method, a train service can be counted as 'punctual' if its arrival lateness (at the terminating station) is less than 5 minutes for London and South East and regional services or 10 minutes for long distance services. A train is counted as ‘right-time’ if its arrival lateness (at the terminating station) is less than 1 minute. And a train is counted as ‘significantly late’ if its arrival lateness (at the terminating station) is no less than 30 minutes. So how to define a ‘successful realisation’ of a given itinerary? A ‘right-time’ arrival (i.e. less than 1 minute late), or a ‘punctual’ arrival (i.e. less than 5 or 10 minutes late), or a ‘not significantly late’ arrival (i.e. less than 30 minutes late)?

Here, in the context of pre-trip information about itinerary-level performance statistics, a ‘not significantly late’ measurement is obviously not as good as a ‘right-time’ or ‘punctual’ measurement. For example, suppose the scheduled arrival time at the destination station of a given critical itinerary is 12:01 p.m. and two itinerary-level performance statistics have been calculated: one is 60% based on a 5-minutes threshold and the other is 95% based on a 30-minutes threshold. And these two pieces of performance information are respectively delivered to two different travellers. Which piece of information would be more useful? The answer is it is perhaps the statistic calculated from the 5-minutes threshold. This is because the 5-minutes statistic provides an information receiver with a considerably small ‘uncertainty interval’ (i.e. [12:01, 12:05]), which makes the information receiver easy to arrange his/her subsequent activities at the destination. By contrast, a wide spectrum of possible values (i.e. [12:01, 12:30] based on the 30-minutes statistic) could bring difficulty in planning subsequent activities. Therefore, either a single TAL is adopted of 1 minute, 5 minutes, or 10 minutes, or multiple TALs are adopted (e.g. 1 minute and 5 minutes) and respectively computed in the algorithm.

4.3.7 The treatment of train cancellations

Once the involved NTTs, AWs, and TALs are determined, Algorithm 3 can then be executed to generate a list of itinerary-oriented performance statistics corresponding to the recommended list of critical itineraries that follow a given critical route (see Tables 4.2 and 4.3 for illustrations). However, there exists a potential controversy over the treatment of cancelled trains: should predictable cancellations be taken into account when computing itinerary-level performance statistics? Tables 4.2 and 4.3 below provide a realistic example that may help better understand this issue.

Table 4.2 and Table 4.3 respectively present a list of itinerary-oriented performance statistics (IPSs) differing only in the treatment of predictable cancellations: while Table 4.2 is obtained from a sample including predictable cancellations, Table 4.3 does not take into account those predictable cancellations. The studied route is Bournemouth – Southampton Central – Brighton, and the observation period is between 20 July 2015 and 13 September 2015 (8 weeks in total). Moreover, a route-specific NTT (net transfer time) of 1 minute is adopted, and the threshold for arrival lateness (TAL) is set to 5 minutes.

Table 4.2 Itinerary-oriented performance statistics Example One: predictable cancellations included

Dep.	Arr.	Dur.	Chg.	IPS (%)
09:55	12:18	2h23m	1	90
10:59	13:18	2h19m	1	79
11:59	14:18	2h19m	1	87
12:59	15:18	2h19m	1	85
13:59	16:14	2h15m	1	28
14:59	17:18	2h19m	1	76
15:59	18:18	2h19m	1	64
16:59	19:19	2h20m	1	54
17:59	20:18	2h19m	1	74
18:59	21:18	2h19m	1	79

NOTE: Bournemouth → Southampton Central → Brighton, between 20 July 2015 and 13 September 2015 (8 weeks), NTT = 1 minute, TAL = 5 minutes.

Table 4.3 Itinerary-oriented performance statistics Example Two: predictable cancellations excluded

Dep.	Arr.	Dur.	Chg.	IPS (%)
09:55	12:18	2h23m	1	90
10:59	13:18	2h19m	1	79
11:59	14:18	2h19m	1	87
12:59	15:18	2h19m	1	85
13:59	16:14	2h15m	1	58
14:59	17:18	2h19m	1	76
15:59	18:18	2h19m	1	64
16:59	19:19	2h20m	1	54
17:59	20:18	2h19m	1	74
18:59	21:18	2h19m	1	79

NOTE: Bournemouth → Southampton Central → Brighton, between 20 July 2015 and 13 September 2015 (8 weeks), NTT = 1 minute, TAL = 5 minutes.

It can be seen from these two tables that the only difference lies in the IPS that corresponds to the (recommended) itinerary with a scheduled departure time of 13:59: while the associated IPS is 28% in Table 4.2, this value becomes 58% in Table 4.3. At first glance, both of these two statistics (i.e. 28% and 58%) seem counter-intuitive: why should an off-peak earlyafternoon journey have such a poor performance in terms of punctuality and reliability? After a close examination of the historical data, it is recognised that this 'anomaly' can largely be attributed to a major rail strike by First Great Western[11] staff during August 2015. But why could this itinerary have two quite different versions of IPS? Which one reflects the reality? In order to answer these two questions, it is necessary to have a good understanding of railway planning processes (c.f. Section 2.5).

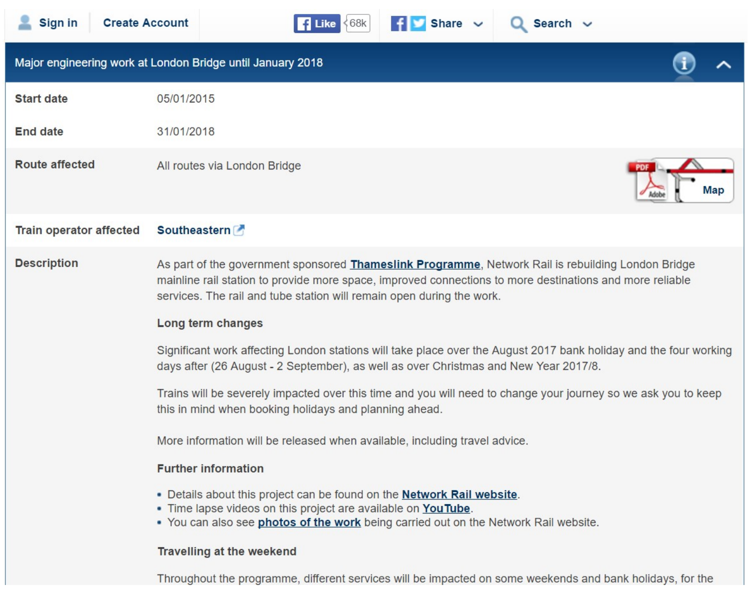


Figure 4.17 An illustration of pre-trip information about major engineering works in the long planning horizon (Source: www.nationalrail.co.uk/, accessed 25 Jan 2017)

Recall that in the introduction to railway planning processes (in Section 2.5), two relevant notions have been briefly mentioned: short-term rescheduling (c.f. Subsection 2.5.1) and daily timetable (c.f. Subsection 2.5.2). A daily timetable is designed and constructed well before (usually several months before) the scheduled train services of a given railway line are put into production on the predetermined date in the tactical planning phase. The introduction of daily timetables in European railways is primarily to better adapt to daily variations in transport demand and infrastructure conditions. For example, a weekday timetable often remains the same for (normal) weekdays during a timetable period (several months or a year), but may be quite different from a weekend/holiday timetable in terms of quantity (e.g. more services on weekdays) and quality (e.g. shorter scheduled travel times on weekdays) of the planned train services. Moreover, major engineering works (see Figure 4.17 for an illustration)

[11] Rebranded as Great Western Railway at the time of writing this thesis.

can also be reflected in daily timetables well in advance. If some of the train services in a weekday timetable are cancelled in a holiday timetable or due to major engineering works (on a future date), these cancelled services are usually recognised as *planned cancellations* (DAB, 2016) due to the fact that they are fully predictable and are reflected in published timetables well in advance (several months or a year before the time of travel).

Although not strictly analogous to the mechanism of daily timetables in the tactical planning phase, short-term rescheduling is widely adopted by rail operators to deal with predictable changes in the published timetables in the operational planning phase. These predictable changes may or may not include those short-term (i.e. a duration of several days or several weeks) events such as rail strikes, local infrastructure improvements, crew shortage, etc. If some of the train services in a published timetable are, either thoroughly or partly, cancelled due to predictable reasons, these cancelled services can be regarded as predictable cancellations. Here, the term ‘predictable’ is used to emphasise that although these cancellations are planned to happen in the short term, they can still be reflected in a revised timetable (and shown in a journey planning system) well in advance (e.g. several days or several weeks ahead). In contrast, if a train service is planned to be cancelled in the very short term (i.e. several hours ahead), it can hardly be reflected in the published timetable and hence is regarded as stochastic/unpredictable.

Different ways of treating predictable cancellations would result in quite different performance statistics. Let us look back at the illustrative example in Tables 4.2 and 4.3. Close scrutiny of the relevant historical train movements data reveals that the connecting train involved in the ‘abnormal’ itinerary (with a scheduled departure time of 13:59) is operated by First Great Western (originating from Great Malvern, calling at Southampton Central, and terminating at Brighton), and it was rescheduled (terminating instead at Bristol Temple Meads and thoroughly cancelled between Bristol and Brighton) between 3 August 2015 and 31 August 2015 (involving 20 weekdays). That is, this 13:59 itinerary has not been successfully realised on each of these 20 weekdays due to the cancellation of the connecting leg. Here, a question arises: should these 20 failures be taken into account in the calculation of its performance statistic?

Two different perspectives can be distinguished in the treatment of these 20 records: 1) they should be taken into account because they could reflect, at least, that the services provided by this operator are not as reliable as those provided by the others; 2) they should not be taken into account because these cancellations are predictable and can be reflected in pre-trip information well in advance, the impact of which on itinerary planning is very limited. Through an inspection of the intermediate results, it is found that there are 11 records in total that are recognised as successful realisations (of this studied itinerary). The sample size for this studied itinerary would be 39 (8 weeks, 40 weekdays, the Summer Bank Holiday is excluded) if the first perspective is adopted, which would result in an IPS (itinerary-oriented performance statistic) of 28% (i.e. 11/39, c.f. Table 4.2). By comparison, the sample size would be 19 (i.e. 39 – 20, those 20 'predictable cancellations' are excluded) if the second perspective is adopted, which would result in an IPS of 58% (i.e. 11/19, c.f. Table 4.3).

So which of the above two perspectives should be adopted? Here, in the context of passenger-oriented itinerary planning, the second perspective is preferred due to the following two reasons. Firstly, current journey planning techniques have long been able to deal with those predictable cancellations by updating the corresponding published timetables. That is, as long as a cancellation can be reflected in the published timetable well before the time of travel, the corresponding service will not be adopted (by journey planning systems) and hence will not enter the recommended itinerary list for a given query. Therefore, the impact of such cancellations on the pre-planning of a given journey would be trivial. Secondly, those events that result in such (predictable) cancellations (e.g. major rail strikes in the above-mentioned example) are quite rare in reality, and passengers would not encounter such events in most cases during a given observation period (e.g. a period of several months). In fact, these rare scenarios can be regarded as outliers in the sense of statistical analysis.

4.3.8 Modifying advertised arrival times

In the previous subsections (4.3.3 – 4.3.7), the core part (i.e. Steps 1 – 3) of Algorithm 3 has been explained in detail: in the next step, Algorithm 3 can be fully implemented by adding a user-friendly interface, as long as there are no further non-technical factors restricting the deployment of such a travel information tool. Comparing between Algorithm 3 and Algorithm 4 (c.f. Subsection 4.3.2), it can be found that although the first two steps (corresponding to data pre-processing) are the same, divergences turn up from Step 3 onwards. The main objective of this subsection is hence to illustrate/clarify the technicalities involved in Steps 3 – 5 of Algorithm 4.

In general, Algorithm 3 can be viewed as an augmented version of those individual-leg-oriented performance statistics (c.f. Subsection 4.2.7), which is designed to enable uncertainty-aware journey planning. Compared with those individual-leg-oriented performance statistics (see Figures 4.6 and 4.7 in Subsection 4.2.7), the output of Algorithm 3 (i.e. itinerary-oriented performance statistics, see Table 4.3 for an illustration) does not require that a passenger must have sufficient experience in rail travel or must be good at mathematics so that he/she could estimate the overall performance of a given itinerary plan by integrating fragmented information (about each involved service leg) by themselves. In realistic applications, however, itinerary-oriented performance statistics (generated by Algorithm 3) may still cause inconvenience to information consumers (passengers), despite their advantage over individual-leg-oriented performance statistics. For example, we can learn from Table 4.3 that the probability of a successful realisation of the 13:59 itinerary (i.e. arriving at Brighton before 16:19 (16:14 + 5)) is 58% during the 8-weeks observation period. But this additional performance information may still be insufficient for decision making (i.e. whether to choose this itinerary on a future date[12]): how much delay would be expected in the other 42% unrealised situations? For some of the relevant passengers, this additional statistic provided (i.e. 58%) may increase their anxiety about being exposed to huge uncertainty, rather than helps them make better decisions.

To ameliorate the potential uneasiness resulting from a feeling of being gambling, an alternative approach is proposed: performance-based pre-modification of advertised arrival times (Algorithm 4, c.f. Subsection 4.3.2). The mechanism of performance-based pre-modification of advertised arrival times (PBPM) can be roughly described as follows: for a given recommended itinerary, its scheduled travel time and scheduled arrival time at its destination station are modified well before the time of travel, based on its overall performance in the last several weeks.

The general idea of PBPM (Algorithm 4) does not come out of nowhere: it has been inspired by the existing functionality of service-specific modification of arrival/departure times in the domain of real-time passenger information. Figures 4.18 and 4.19 below provide an illustration of real-time delay information about modified (advertised) arrival/departure times.

[12] The reference point is 13 September 2015.

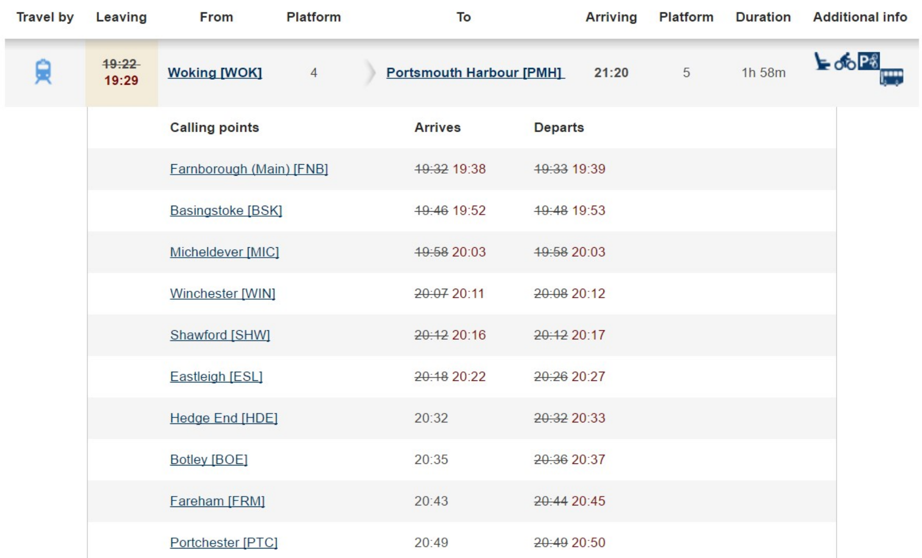


Figure 4.18 Live information about modified arrival/departure times Example One: NRE website (Source: www.nationalrail.co.uk/, accessed 19:20, 22 Jan 2017)

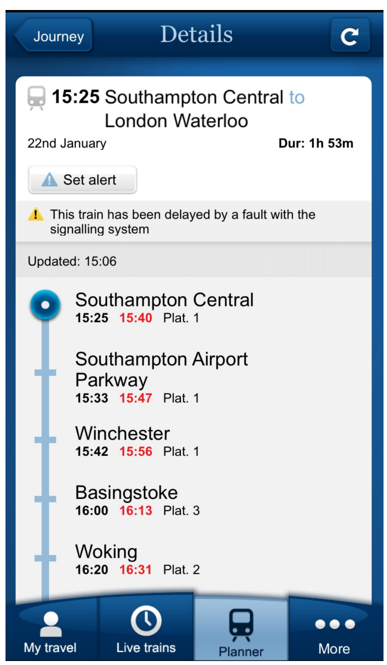


Figure 4.19 Live information about modified arrival/departure times Example Two: NRE mobile app (Source: www.apple.com/itunes/, accessed 15:06, 22 Jan 2017)

It can be seen from Figures 4.18 and 4.19 that the mechanism of these real-time updates is to adjust a passenger's expectation of potential delays before or during his/her trip. Intuitively, these real-time updates would be helpful in the sense that they enable passengers to know about and prepare for the oncoming delays and hence may reduce the potential risk (consequence) of these delays. However, these real-time delay updates are often not accessible until it is very near to the time of travel (see Figures 4.18 and 4.19 for example). In such a context, PBPM (Algorithm 4) is proposed, which takes the idea of delay updates one step further: the pre-trip information about potential arrival delays (at the destination station of a given itinerary) would be accessible well in advance (several days before the time of travel) to enable passengers to make better choices by pre-modifying the scheduled/advertised arrival times (and hence the scheduled/advertised journey times).

Let us look back at the technicalities in Algorithm 4 (c.f. Subsection 4.3.2). Steps 1 – 2 are for data pre-processing and are the same with those in Algorithm 3. That is, a route-view timetable (RVT, c.f. Subsection 4.3.4) should firstly be constructed from historical train movements data, and itinerary-specific net transfer times (NTTs) and actual windows (AWs) should be predetermined.

Once these data pre-processing steps are completed, Step 3 is then executed to calculate the probability of no missed transfers (i.e. all the involved transfers are successfully realised) for each studied critical itinerary. Note that this step does not require a predefined TAL (threshold for arrival lateness, required by Step 3 of Algorithm 3) because these calculated probabilities are not used to present heuristic performance measures (c.f. Algorithm 3) but are used instead to pre-modify the scheduled/advertised journey times and arrival times.

Table 4.4 An illustration of Steps 3 – 4 of Algorithm 4

Dep.	**serviceF**	**seviceC**	**successRate (%)**	**averageLatenessC (mins)**
09:55	Y41233	W83537	90	2.05
10:59	Y41237	W83538	95	0.75
11:59	Y41241	W83539	75	0.39
12:59	Y41245	W83540	85	0.32
13:59	Y41250	P01078	100	4.60
14:59	Y41254	W83541	90	3.11
15:59	Y41259	W83542	85	3.83
16:59	Y41263	W83543	95	1.53
17:59	Y41268	W83544	95	3.68
18:59	Y41274	W83545	85	2.88

NOTE: Bournemouth → Southampton Central → Brighton, between 12 Oct 2015 and 6 Nov 2015 (4 weeks)

Step 4 is to compute the average lateness at the destination station for the k^{th} involved leg of each critical itinerary. Here, to ensure generality, it is assumed that there are exactly k trains (service legs) involved in each studied itinerary. That is, this step is to take into account the scenario in which the probability of no missed transfers (i.e. the output of Step 3) is 100% but the final leg involved (i.e. the k^{th} involved leg of a given itinerary) has poor performance in terms of punctuality at the destination station. Table 4.4 gives such an illustration.

This illustrative example is based on a 4-weeks sample (12/10/2015 – 06/11/2015) of the route Bournemouth – Southampton Central – Brighton. Each studied itinerary is represented, for convenience, by its scheduled departure time from Bournemouth (e.g. 09:55 represents the itinerary with a scheduled departure time of 09:55). Each string (e.g. Y41233) under 'serviceF' and 'serviceC' is the service identifier of a particular train: those under 'serviceF' represent feeder trains and those under 'serviceC' represent connecting trains. The values under 'successRate' are obtained from Step 3, each of which represents the probability of no missed transfers for a particular itinerary. Those values under 'averageLatenessC' are calculated from Step 4, each of which represents the average lateness at the destination station (i.e. Brighton in this example) for the k^{th} involved (i.e. the second) leg of a particular itinerary. For example, the value corresponding to the 13:59 itinerary (i.e. 4.60) can be interpreted as the average lateness of the connecting leg (identified by 'P01078') of this itinerary (i.e. 'P01078' arrived, on average, 4.6 minutes later than the scheduled arrival time at Brighton during this 4-weeks observation period). Meanwhile, we can see from this example (i.e. the 13:59 itinerary) that although the transfer involved performed very well (i.e. 100% successful realisation) during this period, the connecting leg (i.e. 'P01078') had poor performance in terms of punctuality at Brighton.

Step 5 is the final step of Algorithm 4, which is to modify the advertised arrival time (and also the advertised journey time) of each studied itinerary based on the historical performance information obtained from Steps 3 – 4. Firstly, it is necessary to explain the relevant notations/symbols in this step: $jt_0(\cdot)$ represents the average journey time without missed transfers (during a given observation period); $jt_1(\cdot)$ represents the average journey time with exactly one missed transfer (among the k-1 involved transfers, based on an assumption of k legs); $arr_s(\cdot)$ means scheduled arrival time (of a particular train); $dep_s(\cdot)$ means scheduled departure time; $arr_m(\cdot)$ means modified arrival time; $p_0(\cdot)$ and $\delta(\cdot)$ represent the statistics obtained from Steps 3 – 4 (c.f. 'successRate' and 'averageLatenessC' in Table 4.4);

$HEADWAY_{avg}$ means the average headway of all the involved connecting legs. Note that the equations in Lines 49 – 51 of Algorithm 4 (c.f. Subsection 4.3.2) are actually a simplification of the following equation (Eq. (1)):

$$arr_m(i_c) = dep_s(T1) + p_0 \cdot jt_0 + p_1 \cdot jt_1 + p_2 \cdot jt_2 + \cdots + p_{k-1} \cdot jt_{k-1} \qquad \text{Eq. (1)}$$

That is, the modified journey time (i.e. $\Sigma p_i \cdot jt_i$) should be the summation of all scenarios of missed transfers. In reality, however, the number of involved transfers along a critical route is often not that large (c.f. the list of critical routes identified in British railways in Section 3.7), and those items behind $p_1 \cdot jt_1$ can often be neglected. Therefore, the equations in Lines 49 – 51 in Algorithm 4 can be used as an approximation to Equation (1) when the number of involved transfers is small.

In order to help better understand the mechanism of those computations in Step 5, an explanation of the underlying data structure is presented below, followed by several numerical examples using realistic data.

Firstly, it is necessary to differentiate between three interrelated concepts: *itinerary template*, *itinerary*, and *reconstructed itinerary*. Figure 4.20 below provides an illustration of their relationships. The two rows I1 and I2 in Figure 4.20(a) are templates for the two itineraries i1 and i2 in Figure 4.20(b). The origin, transfer, and destination stations and the scheduled stop times at the corresponding stations (i.e. t1 ~ t8, in which t1 < t5, t2 < t6, t3 < t7, t4 < t8 and t1 < t2 < t3 < t4, t5 < t6 < t7 < t8) are the same, but there is one major difference between the two: i1 and i2 are expected to happen on a specific day (e.g. 30/08/16), whereas I1 and I2 are not constrained by a specific date and can be thought of as an abstraction of a collection of repeated itineraries during a considerably long period (e.g. several months or even several years). The symbols f1, c1, f2, c2 respectively represent a specific feeder/connecting train, while F1, C1, F2, C2 respectively represent a specific collection of feeder/connecting trains that follow the same daily schedule during a given period of time. Note that I1 and I2 are 'adjacent' (and so do i1 and i2 and i1' and i2'), which means the interval between t4 and t8 is exactly the headway of C. Here, the set F (and the set C) can be thought of as a higher level of abstraction that contains F1 and F2 (C1 and C2). Take Table 4.4 as an example: F = the Weymouth to London Waterloo line, C = the Southampton Central to Brighton line ('P01078' is an exception, which belongs to the Great Malvern to Brighton line); if F1 = 'Y41233', then C1 = 'W83537', F2 = 'Y41237', C2 = 'W83538'.

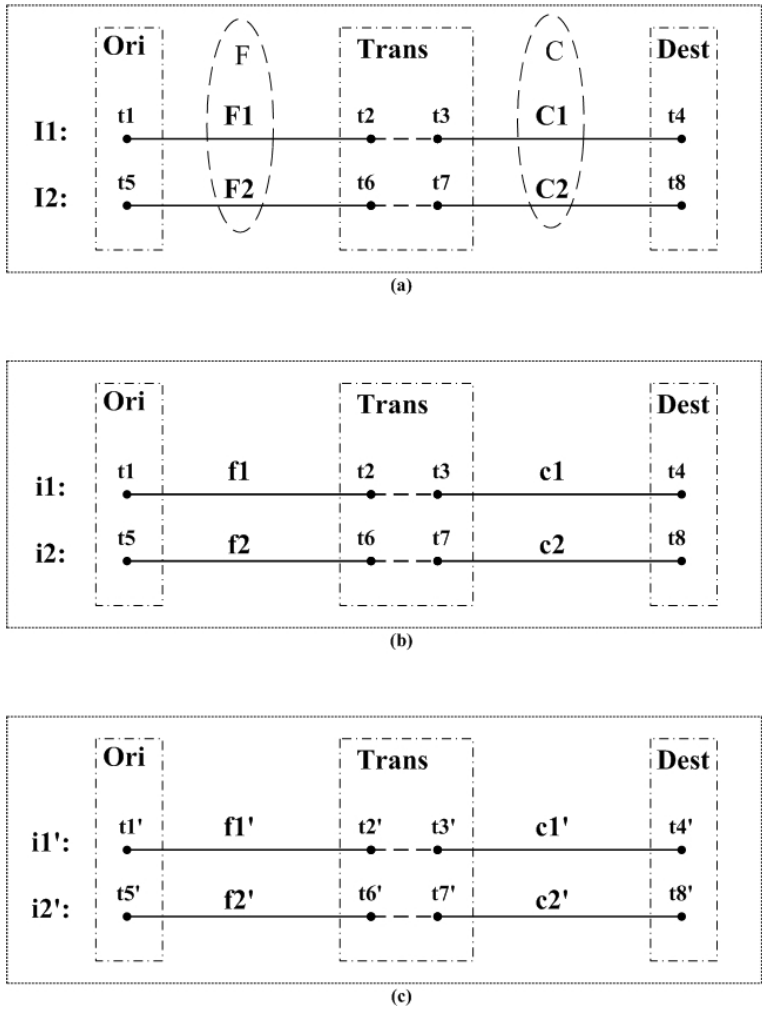


Figure 4.20 An illustration of the relationship between (a) itinerary template, (b) itinerary, and (c) reconstructed itinerary

Figure 4.20(b) and Figure 4.20(c) also have similarities and differences. The origin, transfer, and destination stations are the same; and the corresponding feeder and connecting trains satisfy the following relationships: f1 and f1' belong to F1, c1 and c1' belong to C1, f2 and f2' belong to F2, and c2 and c2' belong to C2. The major difference is that Figure 4.20(b) is a piece of pre-trip information about unrealised journeys, but i1' and i2' in Figure 4.20(c) are reconstructed itineraries that are obtained from splicing the recorded/actual stop times of the corresponding feeder and connecting trains (i.e. f1' + c1', f2' + c2') long after their run date. That is, the run date of the involved trains in Figure 4.20(c) can be thought of as some date before the run date of the involved trains in Figure 4.20(b). Here, the term 'reconstructed' is used to emphasise that the recorded/actual stop times (i.e. t1', t2', … , t8') are not necessarily equal to their counterpart in a planned daily timetable (i.e. t1, t2, … , t8), and that some of the constraints placed on i1 and i2 (e.g. t2 < t3, t6 < t7) do not necessarily hold and the values of t1', t2', … , t8' in i1' and i2' are possible to be invalid (due to train cancellations, c.f. Subsection 4.3.6).

Based on the above Figure 4.20 and Table 4.4, the following equation (i.e. Eq. (2)) can be applied to calculate the modified arrival time (arr_m) for a given itinerary (here, i1 in Figure 4.20(b) is used for illustration):

$$
\begin{aligned}
& arr_m(i1) = \begin{cases} arr_m(I1), arr_m(I1) \geq arr_s(i1) \\ arr_s(i1), arr_m(I1) < arr_s(i1) \end{cases} \\
& where: \\
& arr_s(i1) = arr_s(I1) = t4 \\
& arr_m(I1) = t1 + jt_m(I1) \\
& jt_m(I1) = p_0(I1) \cdot jt_0(I1) + (1 - p_0(I1)) \cdot jt_1(I1) \\
& jt_0(I1) = t4 - t1 + \delta(C1) \\
& jt_1(I1) = t8 - t1 + \delta(C2) \\
& \delta(C1) = E(t4' - t4) = E(t4') - t4 \\
& \delta(C2) = E(t8' - t8) = E(t8') - t8
\end{aligned}
\qquad \text{Eq. (2)}
$$

Most of the symbols in Eq. (2) have been explained earlier in this subsection. Some possibly confusing symbols are those involving 'I1' (e.g. $arr_m(I1)$, $jt_m(I1)$, etc). The purpose of introducing these template-specific symbols is to explain that the algorithm does not differentiate between different weekdays. Here, I1 can be imagined as an abstraction that applies to each weekday during a studied week, and i1 can be imagined as a projection of I1 onto a specific day during the week (i.e. Wednesday). $p_0(I1)$ is the success rate of a realised transfer for I1 during the last several weeks (the reference point is the studied week) and it applies to every 'copy' of I1 (e.g. i1) on every weekday during the studied week. $jt_m(I1)$ is the modified journey time for I1 during the studied week. The calculation of $jt_m(I1)$ takes into account not only the risk of a missed transfer but also the average delay(s) at the destination station (i.e. δ(C1) and δ(C2)). The reason why it is necessary to take into account the average delay(s) at the destination station can be found later in the numerical examples. Note that the calculation of $jt_m(I1)$ only considers at most one missed transfer and has ignored those scenarios under more than one missed transfer (c.f. Eq. (2)). This simplification, however, is unproblematic in the context of British railways because more-than-one-missed-transfer scenarios are very rare (have not been found in the analysis of the current National Rail timetable, c.f. Section 3.7) and can be ignored without loss of precision.

Due to the consideration of $\delta(C1)$ and $\delta(C2)$, the obtained $arr_m(I1)$ is likely to be smaller than the schedule arrival time $arr_s(i1)$. Under this scenario, the modified arrival time $arr_m(i1)$ is no longer $arr_m(I1)$ but is set equal to $arr_s(i1)$ (i.e. no modification under this scenario). This treatment is based mainly on the following two considerations. On the one hand, the ultimate goal of pre-modifying advertised arrival times is to reduce the impact of arrival delays. However, if $arr_m(i1)$ is allowed to be smaller than $arr_s(i1)$, the pre-modification made would increase (rather than reduce) the risk (impact) of arrival delays: an early arrival on average (during a given observation period) may result from the biases from within the adopted sample (e.g. insufficient sample size, seasonal factors, etc). On the other hand, even if the sample adopted is representative (i.e. early arrivals are typical for a studied itinerary during a considerably long period of time), $arr_m(i1)$ should still not be set smaller than $arr_s(i1)$. A slightly earlier arrival (on average) than scheduled may result from the operator's timetabling strategy: running time supplements and/or buffer times may have been incorporated into the published timetables of some routes to offset the impact of potential delays in the tactical planning phase (c.f. Section 2.5 and several real-world examples in the next chapter).

To better understand how Eq. (2) works, three specific calculation examples in Figure 4.21 below are presented. Examples <a> and <b> are based on the real-world data about the route Bournemouth – Southampton Central – Brighton, while Example <c> is based on the real-world data about the route Ilkley – Leeds – Middlesbrough (Note: this route does not belong to the list of identified critical routes in Subsection 3.7.4, but was found to be critical in previous screenings).

runDate	serviceF	serviceC	dep_s_FO	arr_s_CD	late_C	suc_rate	suc_time	fail_rate	fail_time	jt_m	arr_m
09/11/15	Y41259	W83542	959	1098	3.83	0.85	142.83	0.15	201.53	152	1111
09/11/15	Y41263	W83543	1019	1159	1.53	0.95	141.53	0.05	202.68	145	

<a>

runDate	serviceF	serviceC	dep_s_FO	arr_s_CD	late_C	suc_rate	suc_time	fail_rate	fail_time	jt_m	arr_m
09/11/15	Y41250	P01078	839	974	4.60	1	139.60	0	202.11	140	979

<b>

runDate	serviceF	serviceC	dep_s_FO	arr_s_CD	late_C	suc_rate	suc_time	fail_rate	fail_time	jt_m	arr_m
11/07/16	Y15134	Y70644	730	852	-1.55	1	120.45	0	180.78	120	852

<c>

Figure 4.21 Calculation examples of arr_m using real-world data

Compared with Table 4.1 (i.e. the example for Route-View Timetable), only the most relevant columns are extracted to reduce distraction (i.e. the first five columns in the above calculation examples). Columns 6 – 11 (i.e. from 'late_C' to 'jt_m') are auxiliary columns introduced to calculate the last column (arr_m). Here, the first three columns ('runDate', 'serviceF', 'serviceC') are used to uniquely identify a specific itinerary. The two rows (excluding the header row) in Example <a> can be thought of as the two itineraries i1 and i2 in Figure 4.20(b), in which t1 = 959, t4 = 1098, t5 = 1019, t8 = 1159. The values under Columns 6 – 11 correspond to the symbols in the above Equation (2): $\delta(C1) = 3.83$, $\delta(C2) = 1.53$, $p_0(I1) = 0.85$. The other intermediate results can be directly derived from these values. And the obtained result arr_m(i1) = arr_m(I1) = 1111 (since 1111 > 1098(arr_s(i1)), arr_m(i1) = arr_m(I1) = 1111, c.f. Equ. (2)).

Example <b> contains only one row (and so does Example <c>). This is because arr_m(i1) can be calculated without a second row under this scenario (i.e. $p_0(I1) = 1$). The reason why this scenario has been separated from the typical scenario in Example <a> is that it can be used to illustrate the necessity of taking into account the average delay (of the connecting train) at the destination station (i.e. $\delta(C1)$ and $\delta(C2)$ in Equation (2)). As shown in the example, although the transfer can always be successfully realised during the observation period, a punctual arrival at the destination station can still not be expected because the connecting train arrives on average 4.6 mins (rounded up to 5 mins) later than scheduled at the destination station during the observation period. This kind of arrival lateness is also non-negligible (apart from those caused by missed transfers) and ignoring it would affect a correct evaluation of the performance of a studied itinerary.

Example <c> shows how to determine arr_m(i1) in the scenario in which arr_m(I1) < arr_s(i1). Due to the fact that $p_0(I1) = 1$ and $\delta(C1) = -1.55$ (c.f. Example <c> in Figure 4.21), the modified journey time jt_m(I1) = 120.45 ≈ 120 mins and hence the obtained arr_m(I1) = t1 + jt_m(I1) = 730 + 120 = 850, which is less than the scheduled arrival time (arr_s(i1)) of 852. Under this scenario, arr_m(i1) should be set equal to arr_s(i1) (i.e. 852) according to Equation (2).

4.4 Integrating historical performance information into journey planning systems

4.4.1 Presenting independently from journey planning systems

The previous section (i.e. Section 4.3) presents the technical details about the core part of two alternative algorithms that utilise historical train movements data to enhance the pre-trip information about critical routes. Those technicalities are, however, limited to the back end, which is mainly focused on extracting useful information from massive poorly-organised raw data. This subsection and the subsequent subsections are mainly concerned with the front end: how to effectively and efficiently disseminate this additional information about historical performance using existing techniques? Note that since the front end development often requires relatively large capital investments, this section is mainly aimed at presenting the considerations at the technical level, with the aid of an illustrative prototype.

So how to present the additional historical information obtained about the recommended itineraries of a given critical route? A relatively straightforward idea is to refer to the existing 'models' in realistic applications. In the context of Britain's passenger rail system, the most relevant 'models' are perhaps those travel information websites (some of them also have an mobile version) such as Fasteroute Delay Explorer (delayexplorer.fasteroute.com/#/), and My Train Journey (www.mytrainjourney.co.uk/), which are characterised by operating independently from the official information source (i.e. National Rail Enquiries (www.nationalrail.co.uk/) in the context of British railways).

In the context of Algorithms 3 and 4 described in the previous section, this (i.e. mimicking the existing models) means that those itinerary-oriented performance statistics obtained by executing Algorithm 3 (c.f. Table 4.3 for an example) or pre-modified arrival and journey times by executing Algorithm 4 (c.f. Subsection 4.3.8) could be independently published on a self-developed website or mobile application, without interacting with an official journey planning system (e.g. National Rail Enquiries). Although this model (i.e. independently disseminating additional information) could be chosen as the reference point, it suffers from the following three limitations:

- It has quite limited coverage: compared with those long-established travel information sources (e.g. NRE), much less attention would be paid to such newly developed and independently operated travel information sources. That is, most passengers would not utilise (and benefit from) such independent information sources, for it is difficult to make them aware of the existence of such websites (mobile applications).
- Even if passengers are aware of such new information sources, they would still have to integrate different pieces of travel information (from different sources) by themselves. This would cause inconvenience to passengers and push them away from these independent information sources.
- Last but not least, even if the above two potential limitations are set aside, only disseminating (rather than integrating into journey planning algorithms) this additional performance information extracted from historical data could not provide further information about alternative itineraries (journey plans) in the scenario in which poor performance is identified.

The subsequent subsections are to present an alternative approach to disseminating the additional information generated from Algorithm 4 (Note: also applicable to Algorithm 3, but Algorithm 4 is preferred and adopted here, c.f. Subsection 4.3.8 for explanation), with the aid of an illustrative prototype.

4.4.2 Descriptive information (DI) vs. prescriptive information (PI)

A key reason why the benefit is limited of disseminating additional performance information independently from journey planning systems is that these pieces of performance information are largely descriptive rather than prescriptive. According to Ben-Elia et al. (2013), compared with descriptive information (e.g. the average estimated travel times for each route in the context of Ben-Elia et al. (2013)) and post-choice experiential information, prescriptive information about the suggested route has the largest impact on route choice. In the context of presenting the results obtained from Algorithm 4 (and Algorithm 3), this implies that it might be more helpful to provide passengers with additional information about alternative itineraries (to those (already) recommended critical itineraries) than to provide only descriptive information about modified arrival/journey times (or performance statistics).

To help better understand the subtleties of passenger information, Figures 4.22 – 4.25 below provide an illustrative example of the difference between descriptive information and prescriptive information in the context of Algorithm 4 (PBPM, Performance-Based Pre-Modification of advertised arrival times).

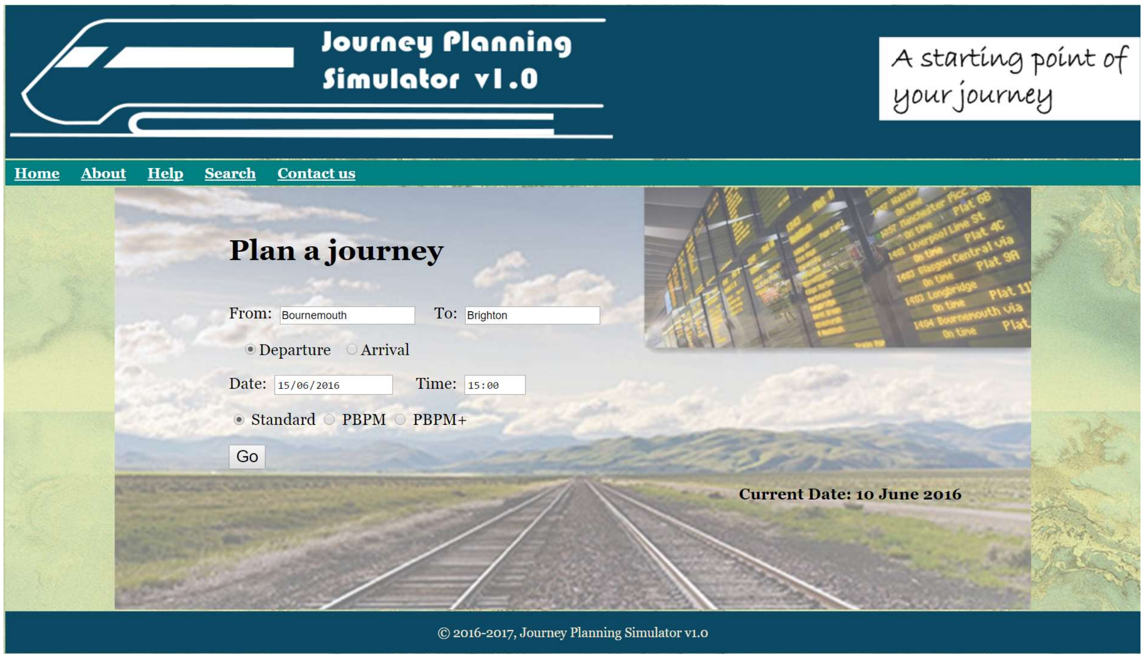


Figure 4.22 A user interface that enables the choice between different underlying algorithms

Figure 4.22 above presents the user interface of the illustrative prototype. It can be seen from the figure that there is no significant difference between this interface and the user interface of those existing journey planning systems (c.f. Sections 4.2 and 4.3). This is not surprising because the input is the same of existing journey planning systems (i.e. a pair of origin and destination stations and the planned departure/arrival date and time), and this prototype is largely an extension/augmentation of existing journey planning algorithms (i.e. it is also built upon existing journey planning algorithms). However, two small distinctions are noteworthy: firstly, an additional functionality is provided to facilitate the transition and comparison between three different modes corresponding to different underlying algorithms (i.e. standard, PBPM, and PBPM+ : further explanation is to be presented later in this subsection). Secondly, it should be noted that although the current date in the figure (i.e. 10 June 2016) is several days away from the planned departure date (i.e. 15 June 2016), the two proposed algorithms PBPM and PBPM+ (to be detailed later in the next subsection) are able to pre-modify the advertised arrival/journey times based on historical performance information, which illustrates their advantage over the existing algorithmic approaches.

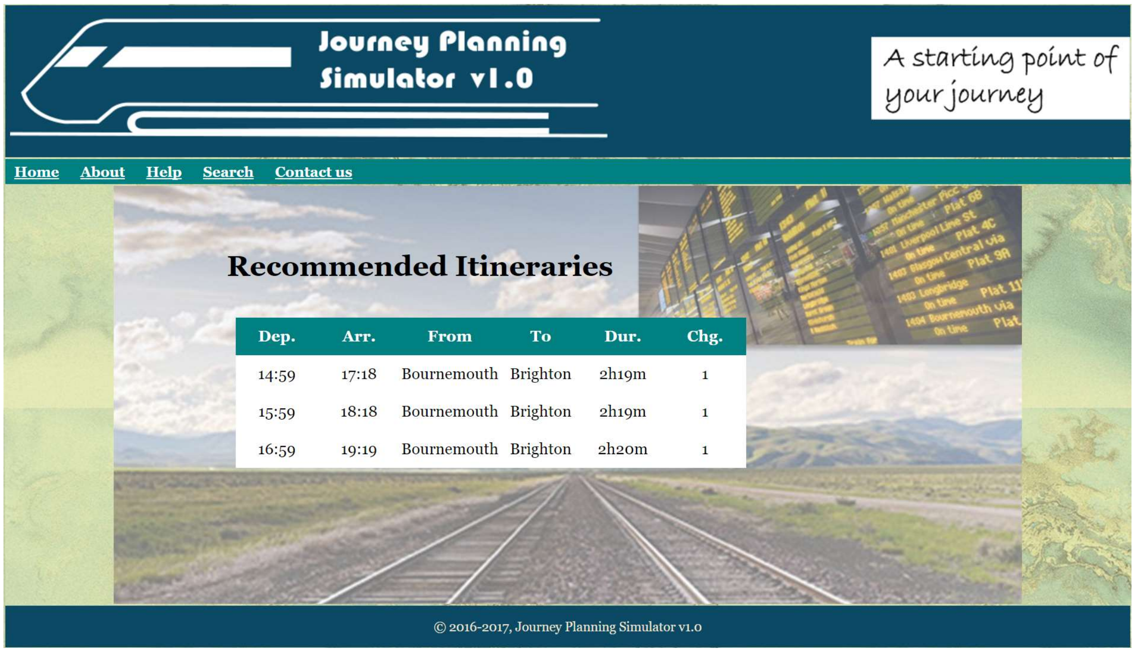


Figure 4.23 DI vs. PI Example One: standard journey planning

Figure 4.23 above displays the result page that corresponds to the ‘standard’ mode. Here, the term ‘standard’ is used to emphasise that the recommended itineraries are calculated purely from the underlying (planned) timetables and historical performance information is not involved. That is, any existing journey planning system would generate the same list of recommended itineraries, as long as the underlying timetables adopted and the relevant parameters chosen (e.g. MTTs, earliest-arrival, etc. c.f. Section 3.4) are the same. Note that the recommended itinerary list in Figure 4.23 is different from the version of National Rail Enquiries (in which additional itineraries transferring at Clapham Junction are also recommended. c.f. Figure 3.7). This difference is due to the difference in the post-processing of the result set (i.e. the recommended itinerary list). An additional filtration rule is added here in the journey planning simulator: in the scenario in which two itineraries I1 and I2 in the result set are non-comparable in the sense of Pareto optimality (c.f. Subsection 2.3.7), if the scheduled travel time of I1 is at least 30 minutes longer than that of I2, then I1 is filtered out. Therefore, those itineraries transferring at Clapham Junction are excluded from the recommended list in Figure 4.23.

From the perspective of information classifications, the recommended itinerary list in Figure 4.23 can be seen as having both the characteristics of descriptive information (DI) and the characteristics of prescriptive information (PI): if a passenger has a relatively flexible schedule (in terms of departure and arrival times), then this piece of information (i.e. the recommended list) would be largely descriptive and he/she would make a choice between the alternative itineraries based on his/her own preferences; conversely, if a passenger has a relatively fixed schedule, then this piece of information would be largely prescriptive because he/she would have no alternative choices. For example, if a passenger has the following flexible schedule: departing no earlier than 14:30 and arriving no later than 18:40, then the recommended itinerary list in Figure 4.20 can be categorised into descriptive information (DI): the passenger can choose between the first itinerary (departing at 14:59) and the second itinerary (departing at 15:59). That is, this piece of information itself could not tell this passenger which itinerary is the best option under this scenario of a flexible schedule. In contrast, if another passenger has the following tighter schedule: departing no earlier than 14:30 and arriving no later than 17:30, then this recommended itinerary list can be categorised into prescriptive information (PI): the passenger would have no alternative choice but the first recommended itinerary (departing at 14:59). That is, this piece of information itself can tell this passenger which itinerary is the best option under such a scenario.

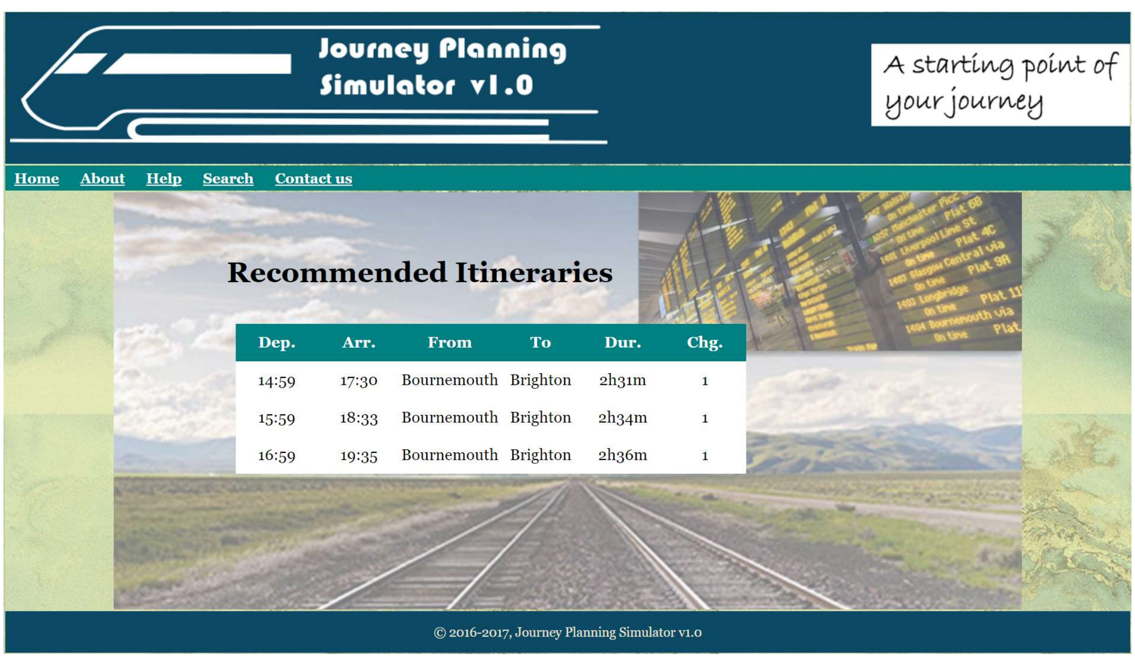


Figure 4.24 DI vs. PI Example Two: additional descriptive information

Figure 4.24 corresponds to the ‘PBPM’ mode in Figure 4.22. ‘PBPM’ is short for ‘Performance-Based Pre-Modification of advertised arrival times’, representing the algorithmic approach described in Algorithm 4 (c.f. Subsection 4.3.2). Comparing Figure 4.24 with Figure 4.23, we can find that the two columns under ‘Arr.’ and ‘Dur.’ are modified: the advertised arrival times of the three recommended itineraries are postponed and the corresponding (advertised) journey times are prolonged, the modifications of which are based on the performance evaluation of the corresponding train services in the previous observation period (c.f. Subsections 4.3.2 and 4.3.8). That is, PBPM (Algorithm 4) exploits both timetable information and historical performance information to enhance the recommended itinerary list obtained from timetable information only (c.f. Figure 4.23).

With respect to information type, the recommended itinerary list in Figure 4.24 inherits from that in Figure 4.23 the characteristics of both DI and PI, and provides those passengers having a relatively tight schedule with additional descriptive information. For example, if a passenger has the following schedule: departing no earlier than 14:30 and arriving no later than 17:20, then this piece of modified information (Figure 4.24) could tell him/her that if he/she chooses the first recommended itinerary, he/she will arrive 10 minutes late on average. By comparison, the timetable information in Figure 4.23 (i.e. the scheduled arrival time of the first recommended itinerary is 17:18) could not make him/her aware of this potential lateness. However, although these pre-modified (recommended) itineraries could make those time-sensitive passengers aware of the potential problems (i.e. delays), they could not provide solutions and hence can be categorised into descriptive information. It is conceivable that the passenger in the above example (in this paragraph) would have no choice but to reschedule the relevant activities (at the origin and/or the destination) or even shift to other modes of transport, which may not be the best result that could have been achieved: on the one hand, although the passenger may benefit from this additional information about potential delays, he/she would still have to take extra time and effort to manually search for alternative plans by himself/herself; on the other hand, although the rail industry may also benefit from this additional information by earning a good reputation for reliable information, certain of its available train services (i.e. capacity) would still be wasted due to the limitations of existing journey planning techniques: a considerable number of available train services between two stations would be filtered out by the underlying algorithms and hence could not be utilised to construct alternative itineraries as necessary (c.f. Section 3.4).

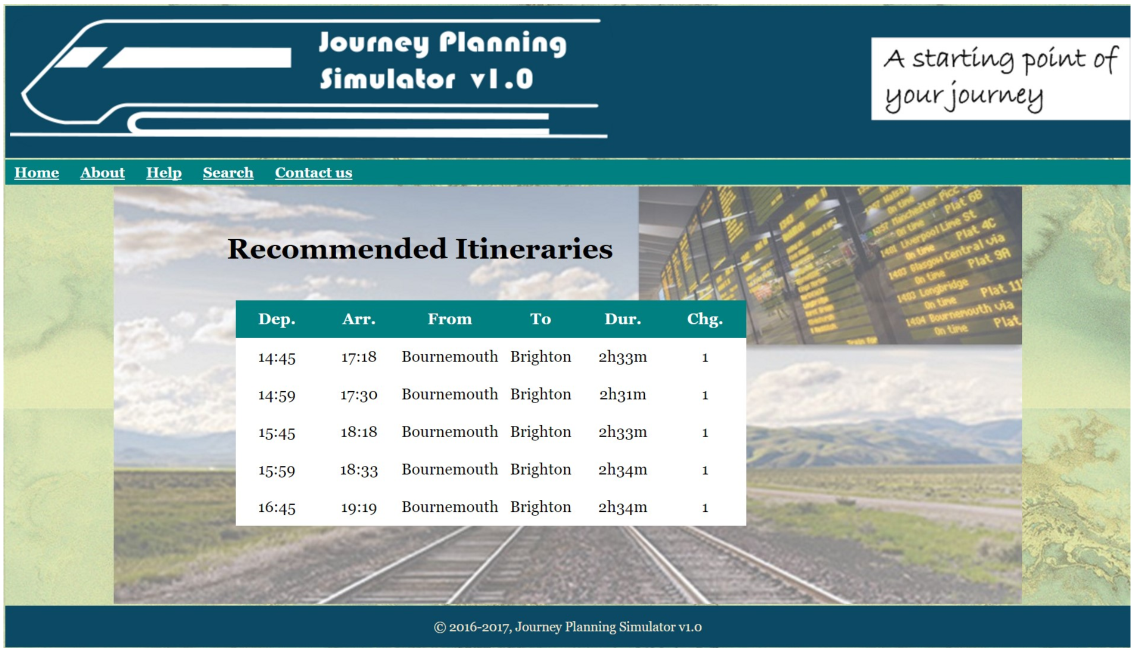


Figure 4.25 DI vs. PI Example Three: additional prescriptive information

A potential solution to the lack of alternative itineraries when necessary is PBPM+ (see Figure 4.25 in the above). PBPM+ can be viewed as an augmented version of PBPM: it not only generates additional descriptive information about potential delays of those (recommended) critical itineraries, but also generates additional prescriptive information about alternative itineraries (to those critical itineraries). That is, PBPM+ tends to be able to cover the most scenarios among the three modes in Figure 4.22 (i.e. standard, PBPM, and PBPM+). Continue the example of the passenger in the previous paragraph (i.e. departing no earlier than 14:30 and arriving no later than 17:20): he/she could adopt an alternative itinerary (to the one departing at 14:59 in Figure 4.24) by departing a little earlier at 14:45 and arriving at 17:18 (i.e. the first itinerary in the recommended list in Figure 4.25). Further details about PBPM+ at the algorithmic level can be found later in the next subsection.

4.4.3 Additional prescriptive information: algorithmic-level considerations

The previous subsection has provided a series of illustrative examples of what an augmented journey planning system would/should be able to do by combining timetable information with historical performance information. Specifically, two proposed algorithmic approaches have been mentioned: PBPM (Performance-Based Pre-Modification of advertised arrival times) and PBPM+. The pseudo code of PBPM has been presented in Subsection 4.3.2, the mechanism of which has been extensively explained in the previous section (i.e. Section 4.3). But the algorithmic-level mechanism of PBPM+ has not been explained. This subsection is to present the algorithmic-level considerations about how to achieve the desired effect in Figure 4.25 in the previous subsection.

Firstly, an explanation of the difference between PBPM and PBPM+ is necessary. By comparing Figure 4.24 with Figure 4.25 (in the previous subsection), we can see that the effect of PBPM+ (c.f. Figure 4.25) is to add into the recommended list (obtained from executing PBPM, c.f. Figure 4.24) additional itineraries. The reason for this difference at the technical level is that PBPM is aimed only at *refining* each critical itinerary in the recommended list by a journey planning system, whereas PBPM+ is aimed at *reconstructing* the recommended list itself. More specifically, PBPM can be viewed as a data mining module functioning independently of a specific journey planning algorithm, but PBPM+ is a combination of PBPM and existing journey planning algorithms. Algorithm 5 below presents the pseudo code for PBPM+.

Algorithm 5: PBPM+

Input: timetable data about a given railway network and historical train movements data about a given critical route

Output: a recommended itinerary list (for the origin-destination pair of the given critical route) in which critical itineraries are refined and alternative itineraries are added

//Step 1: generate a list of refined critical itineraries
run a specific journey planning algorithm (e.g. TDD, TED, RAPTOR, CSA, etc) on the timetable data to generate a recommended itinerary list $\mathbf{RIL}_1$ for the critical route
run PBPM on $\mathbf{RIL}_1$ to obtain a refined itinerary list $\mathbf{RIL}_2$

//Step 2: generate a list of alternative itineraries (to those critical itineraries)
increase the MTT for the transfer station(s) by δ (≥ 1) minutes to obtain a modified MTT_m (= MTT + δ)
rerun the adopted journey planning algorithm on the timetable data adopting MTT_m to generate a recommended itinerary list $\mathbf{RIL}_3$ for the origin-destination pair of the critical route

//Step 3: merge the two lists and refine the resulting list for recommendation
combine $\mathbf{RIL}_2$ with $\mathbf{RIL}_3$ to obtain $\mathbf{RIL}_4$
refine $\mathbf{RIL}_4$ by filtering out those dominated itineraries in the sense of Pareto optimality
return $\mathbf{RIL}_4$
terminate

PBPM+ is generally composed of three major steps. All the notations in bold are two-dimensional tables, and uppercase letters are abbreviations representing either the name of a particular algorithm (e.g. CSA, PBPM, etc) or a constant parameter (e.g. MTT).

PBPM+ consumes both timetable data and historical train movements data of a particular critical route, and yields a recommended itinerary list containing both refined critical itineraries and alternative itineraries to those critical itineraries (see Figure 4.25 in the previous subsection for an illustration).

PBPM+ is created and proposed as an augmented version of PBPM, providing additional pre-trip information about similarly attractive alternatives to those modified critical itineraries (by applying PBPM) as long as such alternatives exist for a given critical route.

The major task of Step 1 is to generate/compute a list of refined critical itineraries. This step can be decomposed into two sub-steps. Firstly, the recommended list of critical itineraries (denoted by RIL_1) needs to be generated by applying a chosen journey planning algorithm (e.g. TDD, TED, RAPTOR, CSA, etc)[13] onto the timetable data. This sub-step can be thought of as the functionality of the 'standard' mode in Figure 4.22 in the previous subsection. Secondly, once RIL_1 is obtained, PBPM (Algorithm 4 in Subsection 4.3.2) can then be applied to refine the critical itineraries in RIL_1 exploiting historical train movements data, the output of which is denoted by RIL_2. This sub-step can be thought of as the functionality of the 'PBPM' mode in Figure 4.22.

Step 2 is mainly aimed at generating/computing a recommended list of alternative itineraries to those recommended critical itineraries. This goal can be achieved by increasing the MTT (minimum transfer time, c.f. Chapter 3) for the station(s)[14] where critical transfers happen, and then (re)running the adopted journey planning algorithm on the timetable data (adopting the modified MTT) to obtain a recommended list of alternative itineraries (denoted by RIL_3). Recall that in the introduction to the existing journey planning algorithms (c.f. Sections 3.4

[13] TDD = Time-Dependent Dijkstra, TED = Time-Expanded Dijkstra, RAPTOR = Round-bAsed Public Transit Optimized Router, CSA = Connection Scan Algorithm. Details about these journey planning algorithms can be found in the previous chapters such as Section 2.3 and Section 3.6.

[14] A critical route may consist of more than one critical transfer, but in the context of this thesis, all the identified critical routes in Britain's passenger rail system involve exactly one critical transfer.

and 3.5), MTTs play an important part in computing a recommended itinerary (for a given query): an MTT is often assigned to each railway station to allow the changeover between different train services. However, due to the impact of delays and disruptions on daily operations, a station-specific MTT tends to be insufficient in the scenario where critical transfers are involved (c.f. Section 3.5). A critical transfer is often associated with a small scheduled window (between the pair of feeder and connecting trains) that barely exceeds the corresponding MTT. For example, the scheduled window for the route London Kings Cross – York – Scarborough is 8 minutes, which is equal to the station-specific MTT for York. And the scheduled window for the route Bournemouth – Southampton Central – Brighton is 4 minutes, which is even less than the station-specific MTT for Southampton Central (i.e. 5 minutes). Note that the route Bournemouth – Southampton Central – Brighton is a special case in which the station-specific MTT (i.e. 5 minutes for Southampton Central) is overlaid with an operator-specific MTT (i.e. 4 minutes between South West Trains and Southern services).

Therefore, by slightly increasing the MTT for the transfer station (and also the operator-specific MTT as necessary) and rerunning the journey planning algorithm, a recommended list (denoted by RIL_3) of alternative itineraries could then be generated (Lines 6 – 7 in Algorithm 5). Here, the obtained alternative itineraries would be the best (apart from those critical itineraries) in the sense of Pareto optimality (in terms of earliest-arrival, number of transfers, journey time, etc). Note that dependent upon the specific parameters adopted (for defining Pareto optimality), the alternative itineraries generated may vary slightly. Continue the example of the route Bournemouth – Southampton Central – Brighton. Both those itineraries transferring at Clapham Junction and those itineraries transferring at Southampton Central and departing (from Bournemouth) at XX:45 (i.e. hourly services at the same time point) would enter the recommended list, if the following rule is added to the definition of Pareto optimality: in the scenario in which two itineraries I1 and I2 in the result set are non-comparable in the sense of Pareto optimality, if the scheduled travel time of I1 is at least 30 minutes longer than that of I2, then filter out I1; otherwise, keep both.

Step 3 is to merge the list of refined critical itineraries (obtained from Step 1) and the list of alternative itineraries (obtained from Step 2) and to refine the combined list. In this step, the operation of merging/combination itself is trivial, involving only some additional sorting of itineraries by scheduled departure time (which is also trivial). Here, the trick lies mainly in the filtration process, which involves a delicate balance between availability of options and conciseness of the result set. Continue the example in the above paragraph. If the same rule is applied to the newly constructed list (i.e. the combined list denoted by RIL_4), the obtained result set would become a little ‘crowded’: each critical itinerary would be associated with two alternative itineraries (i.e. one transferring at Clapham Junction, and the other transferring at Southampton Central). Since those (alternative) itineraries transferring at Clapham Junction have a scheduled travel time of around 2 hours and 55 minutes (c.f. Figure 3.7 in Section 3.4), it might be better to filter out these less efficient and more expensive options to deliver a more concise result set containing only those refined critical itineraries and those XX:45 itineraries (c.f. Figure 4.25 in the previous subsection).

Once the combined list in Step 3 (i.e. RIL_4) has been refined, it can then be disseminated for passenger information. Before finishing the explanation of PBPM+ (i.e. Algorithm 5), the following two points should also be noted.

Firstly, like PBPM (c.f. Algorithm 4 in Subsection 4.3.2), PBPM+ seems computationally intensive but actually is lightweight and would not introduce much extra complexity. The reason for this lies in the following two aspects. On the one hand, either PBPM or PBPM+ can be viewed as a ‘local treatment’ for critical routes only. Recall that the design philosophy of the algorithmic approaches proposed has been described as ‘less is more’: no intervention unless intervention is really necessary (c.f. Subsection 4.3.1). Since only the small set of critical routes (rather than the huge set of all possible routes within a railway network, c.f. Section 3.7) needs to be tackled, the extra computations induced would be trivial. On the other hand, both PBPM and PBPM+ are not truly dynamic: unlike those algorithms designed to be ‘always on-line’ (i.e. constantly update the results; e.g. Müller-Hannemann and Schnee (2009) and Delling et al. (2014a)), PBPM and PBPM+ are designed to be ‘sometimes on-line’ (i.e. update the results on a daily/weekly basis), which significantly reduces the consumption of computing resources.

Secondly, like the limitations of presenting performance-based information independently from journey planning systems (as described previously in Subsection 4.4.1), to what extent PBPM+ would take effect depends on whether it could be adopted and integrated into the official source(s) for rail passenger information. In the context of Britain’s passenger rail system, such an official source is National Rail Enquiries (www.nationalrail.co.uk) operated by Rail Delivery Group[15]. That is, its coverage would be very limited if implemented as an independent travel information website or application. In order to reach a wide audience, it (i.e. PBPM+) need/should be incorporated into some official information source (e.g. National Rail Enquiries).

4.5 Potential limitation

The specific technicalities with illustrations presented in this chapter have shown us how to make full use of those train movements data (available from Britain's rail industry) to generate new information and help enhance the pre-planning of those transfer-involved rail journeys. Despite their considerable potential for practical uses, those publicly accessible rail data about historical train movements should be utilised with caution in scenarios requiring high precision (e.g. microscopic operations analyses). The currently adopted industry standard for data reporting (about train movements) is relatively low, with a precision tolerance of 1 minute (ORDW, 2016b; Network Rail, 2017). Although this level of precision is sufficient in many cases (e.g. real-time delay alerts), it may result in non-negligible errors in more detailed analyses/evaluations requiring high precision.

Figure 4.26 below provides a more concrete context to facilitate the explanation of the relevant issues. This context is a piece of historical data about a recommended itinerary that follows the route Bournemouth – Southampton Central – Brighton, which is extracted from the corresponding Route-View Timetable (c.f. Subsection 4.3.4). The two columns 'serviceF' and 'serviceC' respectively correspond to the service identifiers of the two involved trains for this recommended itinerary, and this pair of train services has happened on 23 Nov 2015 (c.f. the column 'runDate'). The three columns 'stationO', 'stationT', and 'stationD' store the names of the origin station (i.e. BOMO in this example), the transfer station (i.e. SOTON), and the destination station (i.e. BRGHTN), respectively. The three columns 'arr_s_FT', 'dep_s_CT', and 'arr_s_CD' store the values of the scheduled arrival time of the feeder train at the transfer station, the scheduled departure time of the connecting train at the transfer station, and the scheduled arrival time of the connecting train at the destination station, respectively. The three columns 'arr_a_FT', 'dep_a_CT', and 'arr_a_CD' correspond to the

[15] http://www.raildeliverygroup.com/about-us/governance.html, accessed 25 Jan 2017.

recorded (actual) arrival time of the feeder train at the transfer station, the recorded departure time of the connecting train at the transfer station, and the recorded arrival time of the connecting train at the destination station, respectively. Note that those arrival/departure times in the figure have been converted into integers between 0 and 1440 minutes (e.g. 626 = 10:26 a.m.). The four columns ‘display_FO’, ‘display_FT’, ‘display_CT’, and ‘display_CD’ contain information about the status of a given train at a given station, indicating whether (unplanned) cancellations have happened en route (c.f. Subsection 4.3.4 for more details).

runDate	serviceF	serviceC	stationO	stationT	arr_s_FT	arr_a_FT	dep_s_CT	dep_a_CT	display_FO	display_FT	display_CT	stationD	arr_s_CD	arr_a_CD	display_CD
23/11/2015	Y41233	W83537	BOMO	SOTON	626	635	633	635	CALL	CALL	ORIGIN	BRGHTN	738	738	DESTINATION

Figure 4.26 An illustrative example of precision-related issues

From Figure 4.26 we can see that there were no cancellations happening en route (i.e. no status information about ‘CANCELLED_CALL’, c.f. Subsection 4.3.4). Hence, if a net transfer time (NTT) of 1 minute (c.f. Subsection 4.3.5) and a threshold (for arrival lateness, TAL) of 5 minutes are adopted to calculate those itinerary-oriented performance statistics (IPS, c.f. Section 4.3), then the only remaining parameter to determine is the actual window (AW, c.f. Section 4.3) between the feeder train and the connecting train. Since the actual window of this recommended itinerary can be easily calculated (AW = 635 – 635 = 0 minute), this data record will be counted as a failure in calculating a specific IPS.

This level of precision (i.e. integer minutes) can be regarded as acceptable in the context of this thesis because IPS (Itinerary-oriented Performance Measure, c.f. Section 4.3) can be viewed as an extension of PPM (Public Performance Measure, c.f. Network Rail, 2017), which is in essence a heuristic performance measure and represents a rough estimation of an average passenger (c.f. Sections 2.5, 4.3, and 5.2). That is, it would be meaningless to pursue a higher precision of the parameter NTT (e.g. integer seconds) unless the precision of the sample data adopted about historical train movements themselves has been improved.

Consider this particular example of Figure 4.26: it makes no difference whether adopting an NTT (net transfer time) of 56 seconds (c.f. Subsection 4.3.5) or adopting an NTT of 1 minute, because the obtained statistics would be the same unless the precision of the other involved parameter AW (actual window) had been increased (to integer seconds). But suppose that we could get a good estimation of the parameter NTT with a precision of one second (by, for example, long-term field survey) and that the precision of those train movements data had been improved accordingly, then a non-negligible difference would be expected between the statistics calculated under a precision of one minute and their counterparts calculated under a precision of one second. Continue the example of Figure 4.26: this record is counted as a failure (without doubt) under a granularity of one minute, but may be counted as a success under a granularity of one second (e.g. 'arr_a_FT' = 10:34:31, 'dep_a_CT' = 10:35:29, AW = 58 s > NTT = 56 s), which may contribute in the opposite direction to the relevant statistics.

4.6 Conclusions

The pre-trip information about critical routes (if existent) within a railway network would be a potential problem in terms of punctuality and reliability: the recommended itinerary list for a critical route would be full of delay-sensitive transfers, due to the mechanism of existing journey planning algorithms. Theoretically, this problem had better be resolved in the process of timetable design at the tactical planning stage, for the problem could, in essence, be attributed to the underlying timetables (adopted by a journey planning system). In reality, however, timetabling is a complicated process that takes time and is subject to technical feasibility (e.g. the constraint of infrastructure capacity available) and the mediation of the interests of different parties.

A more operable and easier-to-implement approach to improving the pre-trip information (i.e. those recommended itineraries) about critical routes is finding solutions from within the domain of information technology itself to deal with critical routes at the operational planning stage. By reviewing the relevant prototypes in the literature and the relevant applications in the real world, it is recognised that the existing information-related approaches have not truly touched upon the problem of critical routes, either in theory or in practice. But these existing approaches can be utilised as building blocks to develop a solution to the problem of critical routes.

Inspired by some existing travel information technologies, a historical-data-based approach is developed, containing a series of easy-to-implement algorithms. The design philosophy behind the algorithmic approach presented in this chapter is a 'local treatment' of those identified critical routes (rather than a 'holistic treatment' of all possible routes within a railway network), which differs from the various existing approaches. This different treatment could significantly reduce computational complexity and meanwhile avoids distracting information about those non-critical routes.

Three interrelated algorithms are presented and detailed in this chapter, which are named IPS, PBPM, and PBPM+, respectively. IPS (Itinerary-oriented Performance Statistics) has been inspired by those individual-leg-oriented performance statistics accessible from some existing travel information websites. Roughly speaking, IPS can be viewed as an augmented version of those individual-leg-oriented performance statistics: it is designed to compute and present performance statistics that are oriented toward a whole journey (itinerary) rather than toward individual service legs, which would be able to set the information consumers (i.e. rail passengers) free from reprocessing the fragmented information (about individual legs) by themselves.

Despite their advantage over individual-leg-oriented performance statistics, itinerary-oriented performance statistics may still make information consumers feel like they are gambling and hence cause unnecessary inconvenience/uneasiness to them. Based on such a consideration, PBPM (Performance-Based Pre-Modification of advertised arrival times) is developed. PBPM has been inspired by the relevant technologies in real-time delay information: it deserts the use of performance statistics as the 'final products'; instead, it consumes performance statistics as intermediate results to compute the final results (i.e. pre-modified arrival times and journey times) well before the time of travel. Roughly speaking, a pre-modified (advertised) arrival time of a given critical itinerary reflects the 'average lateness' of this itinerary over the last several weeks, incorporating both the risk of missed transfers (reliability) and the average delay at the destination station (punctuality).

Although the final results of PBPM can be readily delivered to end users (passengers) for enhanced pre-trip information, these results (i.e. pre-modified arrival and journey times) are still largely descriptive: for those passengers having a relatively tight schedule, they would still have no alternative choices when the available options (i.e. recommended itineraries) are found to be undesirable. Based on such a consideration, PBPM+ is developed, the purpose of which is to further extend the functionality of PBPM to generate additional prescriptive information about alternative itineraries when necessary. Roughly speaking, PBPM+

incorporates the results obtain from PBPM into existing journey planning algorithms to influence journey planning results. More specifically, this can be achieved by modifying the relevant parameters of a journey planning algorithm and adding to the algorithm additional post-processing procedures.

In the explanation of the three algorithms, open data from Britain's rail industry (i.e. timetable data and historical train movements data) have been extensively exploited to illustrate the data structures adopted, the specific methods employed to determine a series of key parameters (e.g. net transfer time, threshold for arrival lateness, etc), and the considerations about how to present the obtained results. These illustrations can be viewed as a preliminary step in the investigation into the massive and highly detailed rail data available from the Internet. Moreover, the detailed explanation of several introduced concepts/parameters (e.g. net transfer time, predictable cancellation, etc) provides a reference for further refinement of existing journey planning algorithms.

Chapter 5

Quantifying the effect of modified pre-trip information using route-level measures

5.1 Introduction

Chapters 3 and 4 have provided a detailed description of what Critical Routes are, how to efficiently find them out in a given railway network, and how to deal with them using information-related approaches. Two natural questions then arise: Is a piece of modified pre-trip information (resulting from the algorithmic approaches proposed in Chapter 4) really better than its unmodified counterpart (i.e. the version obtained from timetable information only)? If it is, then how much better would be expected? The answer to these two questions may vary if no specific criterion is adopted. This chapter hence tries to answer the above two questions by developing two novel route-oriented measures/criteria.

The main body of this chapter is organised as follows. Firstly, Section 5.2 introduces an absolute measure named Route-oriented Performance Measure (RPM), which can be viewed as an extension of Public Performance Measure (PPM, c.f. Network Rail (2017)). RPM could not only enable a decision maker (operator/manager) to know about route-specific performance in terms of punctuality and reliability during a given observation period, but also enable the comparison of two different pieces of pre-trip information. Adopting RPM, Section 5.3 then presents the evaluation results obtained from the analyses of a number of critical routes in Britain's passenger rail system. After that, a different route-level measure is introduced in Section 5.4, which is a relative measure and is named Route-oriented Utility Measure (RUM). RUM requires the underlying (planned) timetable be a reference point, and takes into account additional factors (apart from punctuality and reliability) such as trip efficiency. Exploiting RUM, analyses of the critical routes in British railways are conducted in Section 5.5. Based on the empirical results of Sections 5.3 and 5.5, Section 5.6 presents more potential applications of RPM and RUM in the field of railway timetabling and operations. Following that, Section 5.7 points out a potential limitation of the proposed measures and the corresponding analytical methods (i.e. the RPM-based method and the RUM-based method), and proposes with illustrations a conceivable solution to the identified limitation in future research. Section 5.8 concludes this chapter.

5.2 Using Route-oriented Performance Measure (RPM) to quantify the effect of modified pre-trip information

5.2.1 Central idea

Since the potential problem of critical routes is mainly embodied in punctuality and reliability (c.f. Chapters 3 and 4), it is natural to consider adapting/extending some existing measure of punctuality and reliability to develop an appropriate standard/criterion for evaluating and comparing two different pieces of pre-trip information about a given critical route. In the context of Britain's passenger rail system, a natural reference point is PPM (Public Performance Measure, c.f. Network Rail (2017)), which is a network-level heuristic measurement widely adopted by European railways and is often presented as the percentage of trains that run within a predefined level of acceptable deviation (e.g. 5 mins) from the officially published timetable (c.f. Network Rail, 2017).

In the introduction to the algorithm of IPS (Itinerary-oriented Performance Statistics) in Section 4.3, PPM has been adapted to generate itinerary-level performance statistics (in terms of punctuality and reliability, c.f. Subsections 4.3.5 – 4.3.7). Those itinerary-level statistics are, however, only meaningful in the context of personal journey planning, and may be of little value in an overall evaluation of a proposed methodology (i.e. modified pre-trip information). Route-level performance indices may be of more interest to rail operators or investors: How a given critical route performs on the whole (in terms of punctuality and reliability) by adopting the existing (unmodified) pre-trip information? What difference can be made by adopting the proposed methodology?

Based on the above considerations, a route-level performance measure named RPM (Route-oriented Performance Measure) is developed, by extending PPM and IPS. The underlying assumptions employed and the relevant technicalities are to be detailed in subsequent subsections.

5.2.2 Definitions and major assumptions

Definition 5.1 *RPM* is the percentage of recommended itineraries (for a given critical route) that have been successfully realised.

Definition 5.2 A *successfully-realised* recommended itinerary corresponds to an average passenger who has arrived at the destination station within a predefined level of acceptable deviation (e.g. 5 mins) from the advertised arrival time.

The above two interrelated definitions provide a general explanation of the proposed route-level performance measure (i.e. RPM). Despite simple descriptions, several assumptions are implicitly adopted in the above definitions. In the following, the major assumptions employed and the differences and similarities between RPM, PPM, and IPS are to be detailed.

Assumption 5.1 Each of those identified critical routes is 'active': a given critical route (recommended by a journey planning system) would be utilised daily by a number of passengers; and even if the number is not large, it is greater than zero.

Assumption 5.2 Each recommended itinerary (for a given critical route) is treated as equally important in the computation of a specific RPM.

Assumption 5.3 Each recommended itinerary (for a given critical route) can be represented by an average (typical) passenger among those having adopted this recommended itinerary.

Assumption 5.4 The advertised arrival time of a given recommended itinerary is not necessarily equal to the scheduled arrival time in the timetable: it could be pre-modified by adopting, for example, the algorithmic approaches proposed in Chapter 4.

The above four assumptions are the major assumptions that are implicitly involved in the definition of RPM. Assumption 5.1 is the most basic assumption and is employed to emphasise that any evaluation or comparison would become meaningless if there exists no transport demand between a given pair of origin and destination stations. Assumption 5.2 is to emphasise that equal weights should be assigned to the recommended itineraries (for a given critical route) unless sufficient knowledge about the exact distribution of passenger flows is obtained, which can be viewed as an application of the principle of indifference (POI, c.f. Section 2.6). It should be noted that the introduction of Assumptions 5.1 and 5.2 is largely due to the unavailability of detailed information (data) about passenger counts and distributions for transfer-involved routes. Note also that Assumptions 5.1 and 5.2 are also implicitly employed in the definition of PPM (c.f. Network Rail, 2017), and Assumption 5.1 is also implicitly included in the calculation of IPS (c.f. Algorithm 3 in Subsection 4.3.2).

Built on Assumptions 5.1 and 5.2, Assumption 5.3 plays a key role in defining/computing RPM. Comparing the definition of PPM (Network Rail, 2017) with that of RPM, we can see that the most significant difference lies in that the computation of a specific PPM involves only one train per count, whereas computing a specific RPM involves more than one train per count. Therefore, the basic unit of RPM becomes an (recommended) itinerary (rather than a train), which increases the difficulty in determining whether a piece/record of sample data should be counted as ‘success’ or ‘failure’: since the characteristics of passengers vary from person to person, some of the passengers adopting a particular recommended itinerary may have successfully realised the itinerary but the others may have been heavily delayed due to, for example, missed transfer(s). In such a context, an ‘average’ or ‘typical’ passenger needs to be introduced to serve as the standard/reference point for evaluation. A subsequent question then arises: how to define an ‘average’ passenger? A general answer to this question is it can be reasonably parameterised by an in-depth investigation into the available real-world data. The technicalities of determining the relevant parameters are to be explained in subsequent subsections.

Assumption 5.4 should not, strictly speaking, be regarded as an assumption: existing technologies have been able to modify the advertised arrival times when it is near to the time of travel (c.f. Figures 4.15 and 4.16 in Section 4.3). Here, it (Assumption 5.4) is used to emphasise that the advertised arrival times are changeable and a specific RPM can have several different versions when adopting different versions of advertised arrival times. This sets RPM free from the implicit assumption underlying PPM and IPS that the reference point adopted for performance evaluation is fixed and has only one version (i.e. the scheduled arrival times), and enables the comparison between the evaluation results of two different pieces of pre-trip information. To help better understand Assumption 5.4, Figure 5.1 below provides an illustrative example.

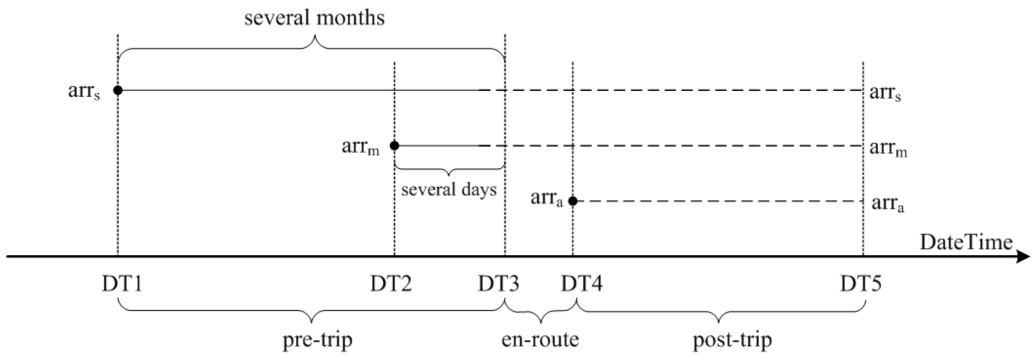


Figure 5.1 An illustration of the different life cycles of arr_s, arr_m, and arr_a. (DT1 = the earliest DateTime the information about arr_s becomes accessible. DT2 = the earliest DateTime the information about arr_m becomes accessible. DT3 = the DateTime the studied journey starts. DT4 = the DateTime the studied journey ends and meanwhile the information about arr_a becomes available. DT5 = the DateTime the evaluation occurs on the condition that arr_s, arr_m and arr_a have been recorded/reconstructed. The three solid dots for arr_s, arr_m, and arr_a and the two solid lines on the right hand side of arr_s and arr_m respectively represent the duration of a specific piece of information.)

From Figure 5.1 above we can see that if the algorithmic approach proposed in Chapter 4 (i.e. Algorithm 4 or 5 in Section 4.3) is adopted, there would be at least two versions of the advertised arrival time of a studied journey (itinerary) in the past: one is the scheduled arrival time in the long-term timetable (denoted by arr_s) that could have been accessible several months before the journey started; the other is the pre-modified (advertised) arrival time (denoted by arr_m) that could have been accessible several days before the journey started. Moreover, additional versions may exist if those real-time updates (not annotated in Figure 5.1, embodied by the dotted part of the two solid lines (corresponding to arr_s and arr_m) between DT2 and DT3) are taken into consideration.

Since real-time updates are transient and are often not recorded in the available historical train movements data, these versions can be ignored in performance evaluation. If we further assume that an average passenger would plan a long-distance and transfer-involved journey (corresponding to a given critical route) several days before the time of travel (i.e. neither too early nor too late), then two versions of RPM could be obtained for a given critical route during a given observation period: one is calculated from a sample set in which each studied itinerary adopts the scheduled (unmodified) arrival time (i.e. arr_s); the other is based on a sample set in which each studied itinerary adopts the pre-modified (advertised) arrival time (i.e. arr_m). Once these two versions of RPM are available, they can then be utilised to conduct quantitative analysis of the effect of modified pre-trip information (about the studied route).

Looking back at Figure 5.1, the only remaining trick lies in the determination/reconstruction of the actual arrival time (denoted by arr_a) of each studied journey (itinerary) in a sample, the technicalities of which are to be explained later in Subsection 5.2.5.

5.2.3 NTT, AW, and TAL

Analogous to the computation of IPS (Itinerary-oriented Performance Statistics, c.f. Section 4.3), three relevant parameters are involved in the computation of RPM: NTT (net transfer time), AW (the actual window between a pair of feeder and connection trains), and TAL (threshold for arrival lateness). Since the technicalities of how to determine these three parameters have been explained in Section 4.3, only several key points to which special attention should be paid are presented here:

- The computation of IPS is oriented to each specific itinerary, whereas the calculation of RPM does not distinguish between different hours of a day: based on Assumption 5.2, all recommended itineraries for a given critical route would be taken into account when calculating a specific RPM during a given observation period.
- Each specific IPS for a given itinerary has only one version, while each specific RPM for a given route can have several different versions (during a given observation period): as has been illustrated in the previous subsection, the available historical train movements data enable us to generate different versions of RPM for a given critical route by adopting different versions of advertised arrival times.
- A specific NTT adopted in an evaluation can be viewed as the amount of time an average passenger needs to complete the transfer, rather than 'the physically possible minimum time required' to walk from the feeder train to the connecting train (c.f. Section 3.5): as has been explained previously in Subsection 4.3.5, allowances have been implicitly included into each adopted NTT in the rounding process.
- The AW(s) for the transfer(s) involved in a sample itinerary can be calculated directly from historical train movements data: as has been illustrated in Subsections 4.3.4 and

4.3.6, the information is available in an RVT (Route-View Timetable) about the actual/recorded arrival time of a feeder train at a transfer station and the actual/recorded departure time of the corresponding connecting train at the same station. Note that train cancellations also influence the determination of an AW, which should also be taken into account (c.f. Subsections 4.3.2 and 4.3.6).

- TAL(s) can be determined by referring to the industry standard: as mentioned earlier in Section 4.3, the industry standard is 5 minutes for commuter or regional services, or 10 minutes for long distance services in British railways. In the analyses of several identified critical routes in Britain's passenger rail system using RPM (to be presented later in Section 5.3), RPMs under 5-minutes TAL and 10-minutes TAL are separately calculated for comparison.
- The influence of predictable cancellations on the evaluation results needs to be taken into consideration: the reason has been explained by an illustrative example in Subsection 4.3.7. Here, in the calculation of RPMs for the identified critical routes in British railways (c.f. Section 5.3), predictable cancellations have been excluded from the adopted sample data.

5.2.4 Sampling issues

Similar to PPM (c.f. Network Rail, 2017) or IPS (c.f. Section 4.3), RPM (Route-oriented Performance Measure) is in essence a statistical concept, the calculation of which is heavily dependent upon the specific sampling method adopted. In previous chapters (specifically, Section 2.4 and Section 4.3), some general sampling-related issues (e.g. sample size) have been touched upon. In this subsection, the considerations about more specific sampling issues are to be presented.

Firstly, the determination of an appropriate sample size is always an unavoidable question. Here, in the context of using RPM to quantify the effect of modified pre-trip information, the issue of determining sample size is twofold: on the one hand, the sample size for calculating an RPM itself should be determined; on the other hand, the sample size for generating the arr_m (pre-modified (advertised) arrival time, c.f. Figure 5.1) of each sample itinerary should also be determined.

As to the calculation of RPMs themselves, a 4-weeks sample size is adopted in the analyses of critical routes in British railways (c.f. Section 5.3), which is based on the following considerations: on the one hand, demand fluctuations (between different months of a year) and seasonal factors (e.g. temperature, humidity, etc) may exert influence on the performance of a studied route and hence need to be controlled; on the other hand, the industry standard in British railways (i.e. PPM, c.f. Network Rail, 2017) also adopts a 4-weeks sample size, which can be viewed as a reference point. Moreover, from the perspective of the number of observations, a 4-weeks sample set of a studied route (in the next section) normally contains more than 100 effective records, which can be regarded as generally sufficient to make those undetected outliers cancel each other out.

As to the computation of the intermediate results (i.e. the arr_m of each involved itinerary), a sample size of 4 weeks is also adopted. At first glance, a 4-weeks sample size seems not large enough in this context: for a specific recommended itinerary, a 4-weeks sample would contain only around 20 records (i.e. 4 weeks, 5 weekdays per week), which is relatively small in the statistical sense. In order to understand why a sample size of 4 weeks is adopted here, it would be helpful to firstly know about the mechanism of arr_m.

According to the algorithm of PBPM (Performance-Based Pre-Modification of advertised arrival times, c.f. Subsection 4.3.2) or PBPM+ (c.f. Subsection 4.4.3), a pre-modified (advertised) arrival time (i.e. arr_m) would be accessible well before (several days before, c.f. Figure 5.1) the time of travel to enable the relevant passengers to have sufficient time to prepare for the potential delays. That is, arr_m can be viewed as an estimation/prediction of the actual/realised arrival time. In order to obtain a good estimation, an assumption of '*the nearer, the more similar*' is implicitly involved in the calculation of those pre-modified arrival times, which is adopted in the analyses in the subsequent section. Roughly speaking, the sample data are updated weekly to generate estimations (i.e. those pre-modified arrival times) for the following week. Figure 5.2 below gives an illustration of the sampling method adopted.

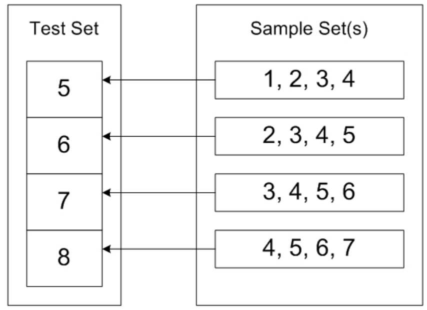


Figure 5.2 An illustration of the sampling method adopted to calculate $arr_m(s)$

Suppose there are 8-weeks historical (train movements) data numbered 1, 2, ... ,8 in chronological order (see Figure 5.2 above). In order to generate/restore the modified pre-trip information (i.e. $arr_m(s)$) on a particular date during this 8-weeks period, those historical data recorded before this date could be utilised as a sample. If a sample size of 4 weeks is adopted, then the 8-weeks historical data can be subdivided into two sets: a test set containing data from Week 5 to Week 8, and a 'dynamic' sample set. That is, some of the data in the test set also serve as a part of the sample set to guarantee that those pre-modified arrival times are always calculated from the most recent 4-weeks sample, the strategy of which is called 'the nearer, the more similar'. Note that the specific technicalities in calculating pre-modified arrival times can be found in the previous chapter (specifically, c.f. Figure 4.17 and Eq. (2) in Subsection 4.3.8).

The sampling method adopted in generating $arr_m(s)$ could also be explained from within the theory of probability and statistics. In the algorithm developed to calculate $arr_m(s)$ (i.e. PBPM, c.f. Subsection 4.3.2), the intermediate result about the 'success rate' of a particular itinerary (i.e. $p_0(i_c)$ in Step 5 of PBPM) can be viewed as the *empirical probability* (or *relative frequency*, c.f. Section 2.4) of a successful realisation of the involved transfer between a pair of feeder and connecting trains. Recall that in the introduction to the fundamentals of probability and statistics (c.f. Section 2.4), the application of *relative frequency* is simultaneously constrained by the law of large numbers (LLN) and repeatability (c.f. Subsection 2.4.3). Here, the sampling method adopted of 'the most recent four weeks historical data' can hence be viewed as a balance between the law of large numbers, the conditions of the trials, and the computational complexity. According to LLN, the sample size should be as large as possible to guarantee the reliability and stability of the empirical probability (i.e. relative frequency). However, a pre-requisite is implicitly involved in the application of LLN: the experiment/trial needs to be repeatedly conducted *under the same (or similar enough) conditions*. In reality (especially in the context of calculating those pre-modified arrival times), a trial (checking whether a specific itinerary is successful on a particular day) is sensitive to a number of external factors such as temperature, humidity, brightness, the characteristics of drivers and equipments, engineering works, seasonal factor (Christmas/Easter/Summer Vacation/School Terms), etc. Therefore, an implicit assumption of 'the nearer, the more similar' is adopted to control the experimental conditions. Note that this assumption is only a general principle behind the sampling method and does not apply to specific data cleanups: any outlier data should be removed from the sample set even if they are temporally near enough to the test data. Further information about the relevant theories of Probability and Statistics can be found in Mood (1974) and Papoulis (1991).

5.2.5 The technicalities in generating reconstructed itineraries

Recall that in Figure 4.17 in Subsection 4.3.8 the difference between three interrelated concepts – itinerary template, itinerary, and reconstructed itinerary – has been briefly explained. However, the technicalities of how to obtain reconstructed itineraries have not been explained. In this subsection, the specific technicalities used in generating reconstructed itineraries are to be illustrated using realistic rail data to help better understand Figure 4.17 and Figure 5.1.

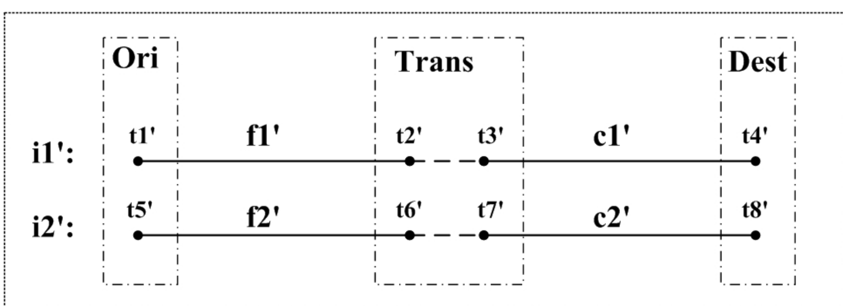


Figure 5.3 An illustrative reconstructed itinerary

Figure 5.3 above is just a copy of Figure 4.17(c): i1' and i2' are two reconstructed itineraries in which f1' and f2' are the feeder trains and c1' and c2' are the corresponding connecting trains; t1' ~ t8' respectively represent the actual/recorded departure time or arrival time at the origin station (denoted by Ori), the transfer station (Trans), or the destination station (Dest).

Moreover, i1’ and i2’ are ‘adjacent’, which means that the interval between t4’ and t8’ is approximately the headway of the connecting leg. Generally speaking, a reconstructed itinerary is obtained from splicing the recorded/actual stop times of the corresponding feeder and connecting trains (e.g. f1’ + c1’) long after the travel date. The term ‘reconstructed’ is used to emphasise that the recorded/actual stop times (i.e. t1’, t2’, … , t8’) are not necessarily equal to their counterpart in a planned daily timetable (i.e. t1, t2, … , t8), and that some of the constraints placed on a planned itinerary (e.g. t2 < t3, t6 < t7) do not necessarily hold and the values of t1’, t2’, … , t8’ in i1’ and i2’ are possible to be invalid (due to, for example, train cancellations).

The value of a reconstructed itinerary is mainly embodied by its arrival time arr_a (c.f. Figure 5.1), which plays an important role in the evaluation of modified pre-trip information. Therefore, the major task involved in reconstructing a particular itinerary lies in the reconstruction of the actual arrival time arr_a. In the following, the considerations about how to reconstruct arr_a and several calculation examples are to be presented.

Firstly, it should be noted that the principle adopted of determining arr_a for each specific itinerary is to maximally simulate/restore how an average passenger would action under a given scenario. With this in mind, the specific technicalities used in reconstructing arr_a(s) are summarised in the decision table (Table 5.1) below.

Table 5.1 The decision table adopted for reconstructing arr_a

Scenario	f1'O	f1'T	c1'T	c1'D	c2'T	c2'D	AW>NTT?	arr_a (i1')
1	√	√	√	√			√	t4'
2	√	√	√	√	√	√	×	t8'
3	√	√	√	√	√	×	×	invalid
4	√	√	√	√	×	√	×	invalid
5	√	√	√	×	√	√		t8'
6	√	√	√	×	√	×		invalid
7	√	√	√	×	×			invalid
8	√	√	×		√	√		t8'
9	√	√	×		√	×		invalid
10	√	√	×		×			invalid
11		×						invalid
12	×							invalid

Following the symbols in Figure 5.3, the actual/reconstructed arrival time of i1’ (denoted by arr_a (i1') in Table 5.1 above) is determined by seven variables: the status of the feeder train f1’ at the origin station (denoted by f1’O: × if ‘CANCELLED_CALL’, √ otherwise), the status of the feeder train f1’ at the transfer station (denoted by f1’T: × if ‘CANCELLED_CALL’, √ otherwise), the status of the connecting train c1’ at the transfer station (denoted by c1’T: × if ‘CANCELLED_CALL’, √ otherwise), the status of the connecting train c1’ at the destination station (denoted by c1’D: × if ‘CANCELLED_CALL’, √ otherwise), the status of the connecting train c2’ at the transfer station (denoted by c2’T: × if ‘CANCELLED_CALL’, √ otherwise), the status of the connecting train c2’ at the destination station (denoted by c2’D: × if ‘CANCELLED_CALL’, √ otherwise), and the indicator of whether the actual window (AW = t3’- t2’) is larger than the net transfer time NTT (√ = yes, × = no).

The status information about f1’O ~ c2’D can be directly found in those ‘display_XX’ columns in a Route-View Timetable (c.f. Table 4.1 in Subsection 4.3.4), and the information about the actual window (AW) between a pair of feeder and connecting trains can be derived from the relevant columns and be stored in an auxiliary column (i.e. the ‘window_a’ column in Figure 5.4, to be explained later). Note that a blank cell in Table 5.1 denotes that the status of the corresponding variable does not affect the determination of the corresponding arr_a.

The logic behind all the 12 scenarios in Table 5.1 is simple: once a passenger finds that the expected waiting time (either at the origin station or at the transfer station) becomes intolerable or that the expected arrival time at the destination station becomes unreasonable/unacceptable, he/she will abandon the currently chosen itinerary (e.g. shift to another transport mode, shift to another itinerary, or cancel the whole journey); otherwise, he/she will continue the current journey and arrive at the destination station at a reasonable time. This logic is based on the observation that a passenger can always get updated information before boarding (from in-station displays/broadcasting or from mobile Internet) about whether there will be a cancelled call at a given station (e.g. the origin/transfer/destination station). And the passenger can hence utilise this piece of real-time information to update his/her pre-trip knowledge and hence actions based on this information.

Take Scenario 1 and Scenario 12 for example. Why t4' is adopted as the arr_a under Scenario 1? This is based on the consideration that an average passenger would continue using the services of f1' and c1' (c.f. Figure 5.3) as long as there had been no informed cancellations and no missed transfer. Why the arr_a cannot be reconstructed under Scenario 12? This is based on the consideration that passengers could have relatively more flexibility before starting a journey: if a cancellation happened before a given journey started, then the relevant passengers would be able to make quite different choices (e.g. shift to another transport mode, shift to another itinerary, or cancel the whole journey), which renders the assumption of an average passenger (c.f. Assumption 5.3 in Subsection 5.2.2) unreasonable.

To make Table 5.1 more tangible, the figure below (i.e. Figure 5.4) provides several numerical examples. Similar to Figure 4.18 in Subsection 4.3.8, these calculation examples contain only the most relevant columns to the calculation of arr_a to reduce distraction. Here, Example <a> is based on the real-world data about the route Harwich Town – Manningtree – Norwich; Examples <b> and <c> are based on the real-world data about the route London Kings Cross – York – Scarborough; and Example <d> is based on the real-world data about the route Ilkley – Leeds – Middlesbrough.

runDate	serviceF	serviceC	dep_s_FO	display_FO	arr_a_FT	display_FT	dep_a_CT	display_CT	arr_a_CD	display_CD	window_a	arr_a
25/07/16	L26378	L26095	928	ORIGIN	950	DESTINATION	955	CALL	1007	CANCELLED_CALL	5	1095
25/07/16	L26381	L26105	988	ORIGIN	1009	DESTINATION	1019	CALL	1095	DESTINATION	10	

<a>

runDate	serviceF	serviceC	dep_s_FO	display_FO	arr_a_FT	display_FT	dep_a_CT	display_CT	arr_a_CD	display_CD	window_a	arr_a
13/07/16	Y71674	Y70219	870	ORIGIN	1034	CALL	1022	CANCELLED_CALL	1056	CANCELLED_CALL	-12	invalid
13/07/16	Y71749	Y70222	930	ORIGIN	1061	CALL	1069	CANCELLED_CALL	1106	CANCELLED_CALL	8	

<b>

runDate	serviceF	serviceC	dep_s_FO	display_FO	arr_a_FT	display_FT	dep_a_CT	display_CT	arr_a_CD	display_CD	window_a	arr_a
01/08/16	Y71716	Y70205	570	ORIGIN	693	CALL	705	CALL	752	DESTINATION	12	752

<c>

runDate	serviceF	serviceC	dep_s_FO	display_FO	arr_a_FT	display_FT	dep_a_CT	display_CT	arr_a_CD	display_CD	window_a	arr_a
27/07/16	Y15126	Y70624	610	CANCELLED_CALL	637	CANCELLED_CALL	651	CALL	730	DESTINATION	14	invalid

<d>

Figure 5.4 Numerical examples of how to determine arr_a using real-world data

In Example <a>, f1'O ~ c2'D = √ √ √ × √ √ (corresponds to Scenario 5 in Table 5.1). Therefore, the actual arrival time of the first row (excluding the header row) is 1095 (i.e. arr_a(i1') = t8'). In Example <b>, f1'O ~ c2'D = √ √ × × × × (corresponds to Scenario

10 in Table 5.1). Therefore, the actual arrival time of the first row is invalid and this row should be removed from the evaluation table. In Example <c>, f1'O ~ c1'D = √ √ √ √ and AW > NTT (12 > 2). Therefore, the actual arrival time of the first row is 752 (t4'). In Example <d>, f1'O = × and f1'T = × (corresponds to Scenario 11 or Scenario 12 in Table 5.1). Therefore, the actual arrival time of the first row is invalid and this row should be removed from the evaluation table.

Note that not every scenario in Table 5.1 can be encountered in a relatively small set of real-world data. And note also that although arr_a cannot be determined (i.e. those invalid values) under most scenarios in Table 5.1, these scenarios are the minority in reality. The majority of the data records belong to the four scenarios under which arr_a can be determined (either is t4' or t8').

5.3 Analyses of several identified critical routes using RPM

5.3.1 Data preparation

In this section, a number of identified critical routes in Britain's passenger rail system are to be analysed using RPM (Route-oriented Performance Measure) proposed in the previous section (i.e. Section 5.2). The aim of these analyses is twofold: on the one hand, they would enable the relevant rail operators or the infrastructure manager to know about the performance of these critical routes in terms of punctuality and reliability; on the other hand, they would enable the relevant stakeholders to know about the effect of the algorithmic approaches proposed (in Chapter 4) on these critical routes (in terms of punctuality and reliability) through tangible results.

The data adopted to conduct these analyses are a large collection of historical train movements data that have been collected from Realtime Trains (RTT): train movements data about the relevant critical routes have been downloaded every day and stored into separate files during a 18-months period between September 2015 and March 2017. As mentioned previously in Section 3.7, RTT data are derived from Network Rail's TRUST system[16] and

[16] https://en.wikipedia.org/wiki/TRUST

are generally well-structured and easier to process than those poorly-structured raw data (from TRUST). Moreover, the database of RTT has relatively limited storage space and is renewed every seven days. Therefore, RTT data needs to be timely collected before the relevant records disappear.

A total of nine routes are analysed, each of which has been identified as critical during the 18-months period. Specifically, these studied critical routes include:

Bournemouth – Southampton Central – Brighton (denoted by BSB)
Ebbw Vale Town – Cardiff Central – Birmingham New Street (denoted by ECB)
Liverpool Lime Street – Manchester Piccadilly – Doncaster (denoted by LMD)
Rugeley Trent Valley – Birmingham New Street – Hereford (denoted by RBH)
Ilkley – Leeds – Middlesbrough (denoted by ILM)
London Kings Cross – York – Scarborough (denoted by KYS)
Harwich Town – Manningtree – Norwich (denoted by HMN)
Knottingley – Wakefield Kirkgate – Nottingham (denoted by KWN)
Sudbury (Suffolk) – Marks Tey – London Liverpool Street (denoted by SML)

Comparing these nine studied routes with the results listed in Section 3.7, several differences can be found. The reason lies mainly in the following two aspects. Firstly, two of the studied routes here (i.e. Bournemouth – Southampton Central – Brighton, Ebbw Vale Town – Cardiff Central – Birmingham New Street) can be viewed as the child routes of the corresponding critical routes listed in Section 3.7 (i.e. Weymouth – Southampton Central – Brighton, Ebbw Vale Town – Cardiff Central – Nottingham), which also satisfy the definition of a critical route (c.f. Section 3.5). Secondly, several critical routes listed in Section 3.7 are not studied here but the route Ilkley – Leeds – Middlesbrough (not listed in Section 3.7) is studied here. This is mainly due to the periodic changes (i.e. half-yearly in Britain) in the underlying timetables: some critical routes listed in Section 3.7 (e.g. the route Oxford – Reading – Gatwick Airport) are newly identified based on the latest version of the planned timetable (for the period from 11 December 2016 to 20 May 2017) and hence no historical data have been collect about these routes; in contrast, the route Ilkley – Leeds – Middlesbrough was identified as a critical route in previous screenings (using the CRF algorithm presented in Section 3.6) during the 18-months period but is no longer critical under the latest version of planned timetables. Despite the changes in the list of identified critical routes, the historical data collected about the route Ilkley – Leeds – Middlesbrough can also be utilised to conduct analyses for the past observation periods.

Five observation periods are selected to conduct the analyses of these nine routes, each of which contains 2-months (8-weeks) historical data (c.f. Subsection 5.2.4): Period 1 (P1) is between 12 October 2015 and 4 December 2015, Period 2 (P2) is between 25 January 2016 and 18 March 2016, Period 3 (P3) is from 13 June 2016 to 5 August 2016, Period 4 (P4) is from 3 October 2016 to 25 November 2016, and Period 5 (P5) is from 16 January 2017 to 10 March 2017. The choice of these five observation periods is based on the following considerations. On the one hand, although it would be enough to adopt a 4-weeks sample (c.f. Subsection 5.2.4) to calculate an RPM to evaluate the performance of a given route (during the 4-weeks observation period), a comparative analysis between modified and unmodified pre-trip information would require an 8-weeks sample in which the data of the first four weeks are needed for generating/restoring the modified pre-trip information (i.e. those pre-modified arrival times, c.f. Figure 5.2 in Subsection 5.2.4). On the other hand, those trans-period samples (e.g. a sample of December and January or a sample of May and June) should be avoided: since the list of identified critical routes is subject to changes in the planned timetable and the planned timetable is updated every six months (in Britain), a route entering the list during a particular timetable period would be likely to be excluded from the list during the subsequent timetable period (e.g. the aforementioned route Ilkley – Leeds – Middlesbrough). Moreover, the choice of these five observation periods also controls the factor of public holidays (e.g. Christmas and Easter): the number of effective observations would be reduced if an 8-weeks period involving public holidays were adopted, due to the fact that a weekday timetable is often different from a holiday timetable.

For a given observation period (i.e. one of the above-mentioned four observation periods), the historical data about a given route (i.e. one of the aforementioned nine studied routes) are subdivided into a test set and a sample set (c.f. Figure 5.2 in Subsection 5.2.4 for an illustration). The test set is used to calculate two different versions of RPM: one is under the assumption of unmodified pre-trip information (denoted by RPM_s in the analyses presented in the subsequent subsections), and the other is under the assumption of modified pre-trip information (denoted by RPM_p in the analyses presented in the subsequent subsections). The sample set is employed to simulate/restore the modified pre-trip information (i.e. those pre-modified advertised arrival times obtained by applying the algorithm of PBPM or PBPM+

presented in Chapter 4 to the sample set). With respect to the technicalities used in sampling, the method presented in Subsection 5.2.4 is adopted. Roughly speaking, the test set is 'static' while the sample set is 'semi-dynamic': the test set contains data about the second half of a given 8-weeks observation period (i.e. from Week 5 to Week 8, c.f. Figure 5.2 in Subsection 5.2.4), but the sample set is different for each of the four test weeks (i.e. Week 5, Week 6, Week 7, and Week 8) to guarantee that those pre-modified arrival times are always calculated from a sample containing data of the most recent four weeks (relative to a given test week).

5.3.2 Route 1: Bournemouth – Southampton Central – Brighton

The route Bournemouth – Southampton Central – Brighton (denoted by BSB) has been identified as critical for a long time (dating back to May 2015). Historical train movements data about this route have been collected since the beginning of September 2015. Several major characteristics of this route can be seen from those illustrative examples in Section 3.4 and Section 4.3: the list of recommended itineraries (by National Rail Enquiries) for this route is full of delay-sensitive transfers, and the connecting leg (i.e. from Southampton Central to Brighton) has relatively low-frequency services (i.e. hourly). Moreover, the determination of the parameter of net transfer time (NTT) has been detailed in Subsection 4.3.5, and a route-specific NTT of 1 minute (i.e. applicable to each studied critical itinerary) is adopted here in this analysis.

In the analysis/evaluation of this route, 10 critical itineraries are studied with scheduled departure times being 09:55, 10:59, 11:59, 12:59, 13:59, 14:59, 15:59, 16:59, 17:59, and 18:59, respectively. The observation periods adopted (see Table 5.2) are Period 1 (12 October 2015 – 4 December 2015), Period 2 (25 January 2016 – 18 March 2016), and Period 5 (16 January 2017 – 10 March 2017). For each of the three adopted observation periods, around 200 records/observations (i.e. 20 weekdays per period and 10 studied itineraries per day) are analysed. The reason why Period 3 (13 June 2016 – 5 August 2016) and Period 4 (3 October 2016 – 25 November 2016) are not analysed is mainly due to planned/predictable cancellations (of the connecting leg, c.f. Subsection 4.3.7) resulting from major rail strikes (by Southern Railway) during these two periods: Figures 5.5 and 5.6 below provide an illustration of the relevant issues.

nationalrail.co.uk

Amended weekday Southern and Gatwick Express services until further notice

Last updated: 04:34 25/09/2016

Incident created: 05/07/2016 10:18

Route affected

Some Southern routes and Gatwick Express

Map of affected area

Train operator affected

Southern

Gatwick Express

Description

In order to give you more certainty for your journey and enable you to better plan, Southern and Gatwick Express have implemented an amended timetable on Mondays to Fridays.

Until train crew availability returns to normal, a revised timetable will be running on the following routes:

- Gatwick Express - some trains will not run
- Brighton - Southampton Central - most trains will not run. Journeys can be made between these stations, but a change of trains will be required
- Hastings - Ashford International - some peak time services will be replaced by buses
- East Croydon / Clapham Junction - Milton Keynes Central - a reduced service will run

From Monday 26 September, the Brighton - Seaford route will be reinstated with a full service.

Figure 5.5 An illustration of planned cancellations for Southern services during Period 3 (Source: www.nationalrail.co.uk, accessed 25 Sept 2016)

nationalrail.co.uk

Changes to train times

Industrial action affecting Southern services

Last updated: 05:27 04/11/2016

Incident created: 29/10/2016 13:31

Route affected

Various Southern routes

Map of affected area

Train operator affected

Southern

Description

The members of the RMT Union are planning to take strike action on various dates in the upcoming months. These include:

- Friday 4 November and Saturday 5 November
- Tuesday 22 November and Wednesday 23 November
- Tuesday 6 December to Thursday 8 December
- Thursday 22 December to Saturday 24 December
- Saturday 31 December to Monday 2 January

Customer Advice:
Details of the Southern service that will be running on the strike days, including the proposed train and bus timetables, is available here.

Local bus operators that provide a service on the affected routes and details of those that are able to accept your tickets are shown here. On other routes with

Figure 5.6 An illustration of planned cancellations for Southern services during Period 4 (Source: www.nationalrail.co.uk, accessed 04 Nov 2016)

As has been explained in Subsection 4.3.7, those planned or predictable cancellations should be excluded in evaluating the effect of pre-trip information: they could be reflected in a revised timetable well in advance (i.e. at least several days before the time of travel) and would not be shown in the recommended itineraries by a journey planning system.

Based on the technicalities explained above, the evaluation results for this route (using Route-oriented Performance Measure, c.f. Section 5.2) are presented in Table 5.2 below. In Table 5.2, RPM_s represents the calculated RPM (Route-oriented Performance Measure) for a given observation period assuming that the unmodified pre-trip information is adopted about scheduled arrival times; RPM_p represents the obtained RPM for a given observation period assuming that the modified pre-trip information about pre-modified arrival times; $\Delta RPM = RPM_p - RPM_s$, representing the change (in RPM) the modified pre-trip information could have brought. Moreover, the parameter of TAL represents the threshold for arrival lateness adopted. The algorithm employed to generate/restore the modified pre-trip information is PBPM in Subsection 4.3.2. The sampling method explained in Subsection 5.2.4 is adopted in calculating RPM_s, RPM_p, and those pre-modified arrival times. The relevant technicalities used in the reconstruction of the actual arrival times can be found in Subsection 5.2.5.

Table 5.2 The evaluation results for BSB using RPM

	Period 1		Period 2		Period 5	
	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10
RPM_s (%)	72.1	83.7	74.2	84.8	75.3	81.2
RPM_p (%)	85.3	87.9	86.4	87.9	84.9	86.6
ΔRPM (%)	13.2	4.2	12.1	3.0	9.7	5.4

From Table 5.2 above, we can see that the performance (in terms of punctuality and reliability) of this route is generally good during the three observation periods and is generally stable between different periods (see the row starting with 'RPM_s (%)'). Moreover, a non-negligible improvement in RPM would be expected if the modified pre-trip information (generated from the proposed algorithm of PBPM) were adopted (see the row starting with 'ΔRPM (%)').

5.3.3 Route 2: Ebbw Vale Town – Cardiff Central – Birmingham New Street

The route Ebbw Vale Town – Cardiff Central – Birmingham New Street (denoted by ECB) has also been identified as critical for a long time: historical train movements data about this route have been collected since the beginning of September 2015. The feeder leg of this route is an hourly Arriva Trains Wales service from Ebbw Vale Town to Cardiff Central, and the connecting leg is an hourly CrossCountry service from Cardiff Central to Birmingham New Street (terminating at Nottingham). A route-specific NTT (net transfer time) of 2 minutes is adopted in this analysis, based on an in-depth investigation into the layout of the transfer station (i.e. Cardiff Central) and the platform allocation of the involved feeder and connecting trains (at the transfer station).

In the analysis/evaluation of this route, 10 critical itineraries are studied with scheduled departure times being 09:38, 10:37, 11:37, 12:37, 13:37, 14:37, 15:37, 16:37, 17:37, and 18:37, respectively. The observation periods adopted are Period 1 (12 October 2015 – 4 December 2015), Period 2 (25 January 2016 – 18 March 2016), Period 3 (13 June 2016 – 5 August 2016), Period 4 (3 October 2016 – 25 November 2016), and Period 5 (16 January 2017 – 10 March 2017). For each of the five observation periods, around 200 records/observations (i.e. 20 weekdays per period and 10 studied itineraries per day) are analysed.

Based on the data and parameters introduced above, the evaluation results for this route are presented in Table 5.3 below. The notations involved have the same meanings with those in Table 5.2: TAL is the threshold for arrival lateness adopted; RPM_s denotes the performance measure calculated based on the unmodified pre-trip information; RPM_p represents the performance measure calculated based on the modified pre-trip information; and ΔRPM indicates the effect of modified pre-trip information on RPM.

From Table 5.3 we can see that the performance of this route is generally good during Periods 1, 2, 3, and 5, but is relatively poor during Period 4 (see the row starting with 'RPM_s (%)'). Moreover, a non-negligible improvement in RPM could have been obtained for Periods 1, 2, 3, and 5, and a significant improvement in RPM could have been obtained for Period 4, if the modified pre-trip information (generated from the proposed algorithm of PBPM) were adopted (see the row starting with 'ΔRPM (%)').

Table 5.3 The evaluation results for ECB using RPM

	Period 1		Period 2		Period 3		Period 4		Period 5	
	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10
RPM_s (%)	72.0	82.8	75.0	88.3	71.1	84.0	54.9	75.1	83.7	92.6
RPM_p (%)	89.2	92.5	90.3	94.4	87.1	90.7	85.5	88.1	96.3	97.4
ΔRPM (%)	17.2	9.7	15.3	6.1	16.0	6.7	30.6	13.0	12.6	4.7

5.3.4 Route 3: Liverpool Lime Street – Manchester Piccadilly – Doncaster

The route Liverpool Lime Street – Manchester Piccadilly – Doncaster (denoted by LMD) has long been identified as a critical route: historical train movements data about this route have been collected since the beginning of September 2015. The feeder leg of this route is an hourly TransPennine Express service from Liverpool Lime Street to Manchester Piccadilly (terminating at Scarborough), and the connecting leg is an hourly TransPennine Express service from Manchester Piccadilly to Doncaster (originating from Manchester Airport and terminating at Cleethorpes). A route-specific NTT (net transfer time) of 3 minutes is adopted in this analysis, based on an inspection of the layout of the transfer station (i.e. Manchester Piccadilly) and the platform allocation of the involved feeder and connecting trains (at the transfer station).

In the analysis of this route, six critical itineraries are studied with scheduled departure times being 09:22, 10:22, 11:22, 12:22, 13:22, and 14:22, respectively. The observation periods adopted are Period 1 (12 October 2015 – 4 December 2015), Period 2 (25 January 2016 – 18 March 2016), Period 3 (13 June 2016 – 5 August 2016), Period 4 (3 October 2016 – 25 November 2016), and Period 5 (16 January 2017 – 10 March 2017). For each of the five observation periods, around 120 records/observations (i.e. 20 weekdays per period and 6 studied itineraries per day) are analysed.

Based on the data and parameters introduced above, the evaluation results for this route are presented in Table 5.4 below. The notations involved have the same meanings with those in the previous subsection: TAL is the threshold for arrival lateness adopted; RPM_s denotes the performance measure calculated based on the unmodified pre-trip information; RPM_p represents the performance measure calculated based on the modified pre-trip information; and ΔRPM indicates the effect of modified pre-trip information on RPM.

From Table 5.4 we can see that the performance (in terms of punctuality and reliability) of this route is generally undesirable (Period 5 is an exception) compared with the other studied routes (see the row starting with 'RPM_s (%)'). Moreover, a significant improvement in RPM could have been obtained for the five adopted observation periods if the modified pre-trip information (generated from the proposed algorithm of PBPM) were adopted (see the row starting with 'ΔRPM (%)').

Table 5.4 The evaluation results for LMD using RPM

	Period 1		Period 2		Period 3		Period 4		Period 5	
	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10
RPM_s (%)	58.5	76.4	75.0	84.8	55.9	78.0	66.1	83.0	81.2	90.6
RPM_p (%)	86.8	88.7	88.4	92.9	87.3	91.5	84.8	86.6	92.9	94.1
ΔRPM (%)	28.3	12.3	13.4	8.0	31.3	13.6	18.8	3.6	11.8	3.5

5.3.5 Route 4: Rugeley Trent Valley – Birmingham New Street – Hereford

The route Rugeley Trent Valley – Birmingham New Street – Hereford (denoted by RBH) has long been recognised as a critical route: historical train movements data about this route have been collected since the beginning of September 2015. The feeder leg of this route is an hourly London Midland service from Rugeley Trent Valley to Birmingham New Street, and the connecting leg is an hourly London Midland service from Birmingham New Street to Hereford. A route-specific NTT (net transfer time) of 3 minutes is adopted in this analysis, by inspecting the layout of the transfer station (i.e. Birmingham New Street) and the platform allocation of the involved feeder and connecting trains (at the transfer station).

In the analysis of this route, nine critical itineraries are studied with scheduled departure times being 08:41, 09:43, 10:41, 11:41, 12:41, 13:41, 14:41, 15:41, and16:41, respectively. The observation periods adopted are Period 1 (12 October 2015 – 4 December 2015), Period

2 (25 January 2016 – 18 March 2016), and Period 5 (16 January 2017 – 10 March 2017). Period 3 (13 June 2016 – 5 August 2016) and Period 4 (3 October 2016 – 25 November 2016) are not analysed due to the lost data about these two periods in the process of data storage and transfer. For each of the three adopted observation periods (i.e. Periods 1, 2, and 5), around 180 records/observations (i.e. 20 weekdays per period and 9 studied itineraries per day) are analysed.

Based on the data and parameters introduced above, the evaluation results for this route are presented in Table 5.5 below. The notations involved have the same meanings with those in the previous subsection: TAL is the threshold for arrival lateness adopted; RPM_s denotes the performance measure calculated based on the unmodified pre-trip information; RPM_p represents the performance measure calculated based on the modified pre-trip information; and ΔRPM indicates the effect of modified pre-trip information on RPM.

From Table 5.5 we can see that the performance of this route is generally good during the three observation periods (see the row starting with 'RPM_s (%)'). Moreover, a small improvement in RPM could have been obtained for Period 1 if the modified pre-trip information (generated from the proposed algorithm of PBPM) were adopted (see the row starting with 'ΔRPM (%)'). However, no/little change in RPM is observed for Period 2 and Period 5, the reason of which is to be explained later in Subsection 5.3.12 by close scrutiny of the sample data.

Table 5.5 The evaluation results for RBH using RPM

	Period 1		Period 2		Period 5	
	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10
RPM_s (%)	79.2	83.2	91.6	92.1	93.7	94.3
RPM_p (%)	83.8	87.3	91.6	92.1	93.7	95.4
ΔRPM (%)	4.6	4.0	0	0	0	1.1

5.3.6 Route 5: Ilkley – Leeds – Middlesbrough

The route Ilkley – Leeds – Middlesbrough (denoted by ILM) was recognised as a critical route by applying CRF (Critical Route Finder, c.f. Algorithm 2 in Section 3.6) to the two previous versions of the National Rail Timetable during 2016, but does not enter the list of critical routes in the screening (using CRF) of the latest version of the National Rail Timetable (which is valid from 11 December 2016 to 20 May 2017, c.f. Section 3.7). Historical train movements data about this route have been collected between January 2016 and September 2016. The feeder leg of this route is a half-hourly Northern service from Ilkley to Leeds, and the connecting leg is an hourly TransPennine Express service from Leeds to Middlesbrough (originating from Manchester Airport). A route-specific NTT (net transfer time) of 3 minutes is adopted in this analysis, based on an in-depth investigation into the layout of the transfer station (i.e. Leeds) and the platform allocation of the involved feeder and connecting trains (at the transfer station).

In the analysis of this route, five critical itineraries are studied with scheduled departure times being 10:10, 11:10, 12:10, 13:10, and 14:10, respectively. The observation periods adopted are Period 2 (25 January 2016 – 18 March 2016) and Period 3 (13 June 2016 – 5 August 2016). For each of the two adopted observation periods (i.e. Period 2 and Period 3), around 100 records/observations (i.e. 20 weekdays per period and 5 studied itineraries per day) are analysed.

Based on the data and parameters introduced above, the evaluation results for this route are presented in Table 5.6 below. The notations involved have the same meanings with those in the previous subsection: TAL is the threshold for arrival lateness adopted; RPM_s denotes the performance measure calculated based on the unmodified pre-trip information; RPM_p represents the performance measure calculated based on the modified pre-trip information; and ΔRPM indicates the effect of modified pre-trip information on RPM (Route-oriented Performance Measure).

Table 5.6 The evaluation results for ILM using RPM

	Period 2		Period 3	
	TAL=5	TAL=10	TAL=5	TAL=10
RPM_s (%)	88.9	90.9	84.5	89.7
RPM_p (%)	90.9	92.9	89.7	93.8
ΔRPM (%)	2.0	2.0	5.2	4.1

From Table 5.6 we can see that the performance of this route is generally good during the two observation periods (see the row starting with 'RPM_s (%)'). Moreover, a small improvement in RPM could have been obtained for both periods if the modified pre-trip information (generated from the proposed algorithm of PBPM) were adopted (see the row starting with '∆RPM (%)').

5.3.7 Route 6: London Kings Cross – York – Scarborough

The route London Kings Cross – York – Scarborough (denoted by KYS) has been identified as critical since May 2016: historical train movements data about this route have been collected since then. The feeder leg of this route is a half-hourly Virgin Trains East Coast service from London Kings Cross to York (terminating at Edinburgh, Newcastle, etc), and the connecting leg is an hourly TransPennine Express service from York to Scarborough (originating from Liverpool Lime Street). A route-specific NTT (net transfer time) of 1 minute is adopted in this analysis, by scrutinising the layout of the transfer station (i.e. York) and the platform allocation of the involved feeder and connecting trains (at the transfer station).

In the analysis of this route, eight critical itineraries are studied with scheduled departure times being 08:30, 09:30, 10:30, 11:30, 12:30, 13:30, 14:30, and 15:30, respectively. The observation periods adopted are Period 3 (13 June 2016 – 5 August 2016), Period 4 (3 October 2016 – 25 November 2016), and Period 5 (16 January 2017 – 10 March 2017). For each of the three adopted observation periods, around 160 records/observations (i.e. 20 weekdays per period and 8 studied itineraries per day) are analysed.

Based on the data and parameters introduced above, the evaluation results for this route are presented in Table 5.7 below. The notations involved have the same meanings with those in the previous subsection: TAL is the threshold for arrival lateness adopted; RPM_s denotes the performance measure calculated based on the unmodified pre-trip information; RPM_p represents the performance measure calculated based on the modified pre-trip information; and ∆RPM indicates the effect of modified pre-trip information on RPM (Route-oriented Performance Measure).

From Table 5.7 we can see that the performance (in terms of punctuality and reliability) of this route is relatively good during Period 3 and Period 5, but is relatively poor during Period 4 (see the row starting with 'RPM_s (%)'). Moreover, a significant improvement in RPM could have been obtained for Periods 3 and 4 if the modified pre-trip information (generated from the proposed algorithm of PBPM) were adopted, although this improvement in RPM would be relatively small for Period 5 (see the row starting with '∆RPM (%)').

Table 5.7 The evaluation results for KYS using RPM

	Period 3		Period 4		Period 5	
	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10
RPM_s (%)	73.7	82.7	58.2	69.9	84.2	89.0
RPM_p (%)	85.9	87.8	78.4	79.1	87.7	91.1
∆RPM (%)	12.2	5.1	20.3	9.2	3.4	2.1

5.3.8 Route 7: Harwich Town – Manningtree – Norwich

The route Harwich Town – Manningtree – Norwich (denoted by HMN) has been identified as a critical route since May 2016: historical train movements data about this route have been collected since then. The feeder leg of this route is an hourly Greater Anglia service from Harwich Town to Manningtree, and the connecting leg is a half-hourly Greater Anglia service from Manningtree to Norwich (originating from London Liverpool Street). A route-specific NTT (net transfer time) of 1 minute is adopted in this analysis, by inspecting the layout of the transfer station (i.e. Manningtree) and the platform allocation of the involved feeder and connecting trains (at the transfer station).

In the analysis of this route, nine critical itineraries are studied with scheduled departure times being 08:28, 09:28, 10:28, 11:28, 12:28, 13:28, 14:28, 15:28, and 16:28, respectively. The observation periods adopted are Period 3 (13 June 2016 – 5 August 2016), Period 4 (3 October 2016 – 25 November 2016), and Period 5 (16 January 2017 – 10 March 2017). For each of the three adopted observation periods, around 180 records/observations (i.e. 20 weekdays per period and 9 studied itineraries per day) are analysed.

Based on the data and parameters introduced above, the evaluation results for this route are presented in Table 5.8 below. The notations involved have the same meanings with those in the previous subsection: TAL is the threshold for arrival lateness adopted; RPM_s denotes the performance measure calculated based on the unmodified pre-trip information; RPM_p represents the performance measure calculated based on the modified pre-trip information;

and ΔRPM indicates the effect of modified pre-trip information on RPM (Route-oriented Performance Measure).

From Table 5.8 we can see that the performance of this route is generally undesirable during Period 3 and Period 4 (see the row starting with 'RPM_s (%)'). Moreover, a significant improvement in RPM could have been obtained for the three observation periods if the modified pre-trip information (generated from the proposed algorithm of PBPM) were adopted (see the row starting with 'ΔRPM (%)').

Table 5.8 The evaluation results for HMN using RPM

	Period 3		Period 4		Period 5	
	TAL=5	TAL=10	TAL=5	TAL=10	TAL=5	TAL=10
RPM_s (%)	55.3	73.2	57.9	79.2	72.9	86.4
RPM_p (%)	82.1	87.2	81.5	87.1	87.0	89.8
ΔRPM (%)	26.8	14.0	23.6	7.9	14.1	3.4

5.3.9 Route 8: Knottingley – Wakefield Kirkgate – Nottingham

The route Knottingley – Wakefield Kirkgate – Nottingham (denoted by KWN) has been screened out as a critical route since September 2016: historical train movements data about this route have been collected since then. The feeder leg of this route is an hourly Northern service from Knottingley to Wakefield Kirkgate, and the connecting leg is an hourly Northern service from Wakefield Kirkgate to Nottingham (originating from Leeds). A route-specific NTT (net transfer time) of 1 minute is adopted in this analysis, based on an examination of the layout of the transfer station (i.e. Wakefield Kirkgate) and the platform allocation of the involved feeder and connecting trains (at the transfer station).

In the analysis of this route, nine critical itineraries are studied with scheduled departure times being 08:53, 09:53, 10:53, 11:53, 12:53, 13:53, 14:53, 15:53, and 16:53, respectively. The observation periods adopted are Period 4 (3 October 2016 – 25 November 2016) and Period 5 (16 January 2017 – 10 March 2017). For each of the two adopted observation periods, around 180 records/observations (i.e. 20 weekdays and 9 studied itineraries per day) are analysed.

Based on the data and parameters introduced above, the evaluation results for this route are presented in Table 5.9 below. The notations involved have the same meanings with those in the previous subsection: TAL is the threshold for arrival lateness adopted; RPM_s denotes the performance measure calculated based on the unmodified pre-trip information; RPM_p represents the performance measure calculated based on the modified pre-trip information; and ΔRPM indicates the effect of modified pre-trip information on RPM (Route-oriented Performance Measure).

From Table 5.9 we can see that the performance (in terms of punctuality and reliability) of this route is generally good during the two observation periods (see the row starting with 'RPM_s (%)'). Moreover, a moderate improvement in RPM could have been obtained for these two observation periods if the modified pre-trip information (generated from the proposed algorithm of PBPM) were adopted (see the row starting with 'ΔRPM (%)').

Table 5.9 The evaluation results for KWN using RPM

	Period 4		Period 5	
	TAL=5	TAL=10	TAL=5	TAL=10
RPM_s (%)	80.6	87.4	89.9	94.9
RPM_p (%)	86.9	90.3	90.4	96.1
ΔRPM (%)	6.3	2.9	0.6	1.1

5.3.10 Route 9: Sudbury (Suffolk) – Marks Tey – London Liverpool Street

The route Sudbury (Suffolk) – Marks Tey – London Liverpool Street (denoted by SML) has been recognised as critical since September 2016: historical train movements data about this route have been collected since then. The feeder leg of this route is an hourly Greater Anglia service from Sudbury (Suffolk) to Marks Tey, and the connecting leg is a half-hourly Greater Anglia service from Marks Tey to London Liverpool Street (originating from Colchester Town/Ipswich). A route-specific NTT (net transfer time) of 1 minute is adopted in this analysis, based on an investigation into the layout of the transfer station (i.e. Marks Tey) and the platform allocation of the involved feeder and connecting trains (at the transfer station).

In the analysis of this route, seven critical itineraries are studied with scheduled departure times being 09:33, 10:26, 11:26, 12:26, 13:26, 14:26, and 15:26, respectively. The observation periods adopted are Period 4 (3 October 2016 – 25 November 2016) and Period 5 (16 January 2017 – 10 March 2017). For each of these two observation periods, around 140 records/observations (i.e. 20 weekdays and 7 studied itineraries per day) are analysed.

Based on the data and parameters introduced above, the evaluation results for this route are presented in Table 5.10 below. The notations involved have the same meanings with those in the previous subsection: TAL is the threshold for arrival lateness adopted; RPM_s denotes the performance measure calculated based on the unmodified pre-trip information; RPM_p represents the performance measure calculated based on the modified pre-trip information; and ΔRPM indicates the effect of modified pre-trip information on RPM (Route-oriented Performance Measure).

Table 5.10 The evaluation results for SML using RPM

	Period 4		Period 5	
	TAL=5	TAL=10	TAL=5	TAL=10
RPM_s (%)	71.3	82.4	85.6	90.6
RPM_p (%)	83.1	88.2	92.1	93.5
ΔRPM (%)	11.8	5.9	6.5	2.9

From Table 5.10 we can see that the performance of this route is generally good during these two periods (see the row starting with 'RPM_s (%)'). Moreover, a non-negligible improvement in RPM could have been obtained for both periods if the modified pre-trip information (generated from the proposed algorithm of PBPM) were adopted (see the row starting with 'ΔRPM (%)').

5.3.11 A summary of the results with interpretation

Subsections 5.3.2 ~ 5.3.10 have respectively presented the evaluation results for each studied route using RPM. Apart from these route-specific performance statistics, we can also synthesise the relevant statistics to obtain some overall performance statistics (analogous to the calculation of PPM, see Network Rail (2017)). However, compared with those specific performance statistics (based on the planned timetable), we are more interested in the changes the modified pre-trip information could bring to the corresponding RPMs (i.e. ΔRPMs). Hence, the relevant ΔRPMs (presented in the result tables in Subsections 5.3.2 ~ 5.3.10) are summarised in Table 5.11 below to help understand the overall effect of the modified pre-trip information on the nine studied critical routes (in terms of RPM).

Table 5.11 Summary based on a pre-defined 'selection rule' (unit: %)

	P1	P2	P3	P4	P5	E(P)	NOTE
BSB	4.2	3.0			5.4	4.2	d = 96 miles, TAL = 10 mins
ECB	9.7	6.1	6.7	13.0	4.7	8.0	d = 98 miles, TAL = 10 mins
LMD	12.3	8.0	13.6	3.6	3.5	8.2	d = 100 miles, TAL = 10 mins
ILM		2.0	5.2			3.6	d = 67 miles, TAL = 5 mins
RBH	4.6	0.0			0.0	1.5	d = 79 miles, TAL = 5 mins
KYS			5.1	9.2	2.1	5.5	d = 238 miles, TAL = 10 mins
HMN			26.8	23.6	14.1	21.5	d = 74 miles, TAL = 5 mins
KWN				6.3	0.6	3.5	d = 62 miles, TAL = 5 mins
SML				11.8	6.5	9.2	d = 68 miles, TAL = 5 mins
						7.2	

In Table 5.11 above, those three-letter abbreviations in the first (i.e. leftmost) column (e.g. BSB and ECB) denote the studied routes (c.f. Glossary at the beginning of this thesis or Subsection 5.3.1). The column titles P1 ~ P5 in the first row respectively represent the five observation periods (c.f. Glossary or Subsection 5.3.1). The values under P1 ~ P5 respectively represent the ΔRPM (in percentage) for a specific route during a specific observation period (c.f. Subsections 5.3.2 ~ 5.3.10). Since two versions of ΔRPMs have been calculated for each studied route for each relevant observation period (one under TAL = 5 mins and the other under TAL = 10 mins, see Subsections 5.3.2 ~ 5.3.10; TAL is short for Threshold for Arrival Lateness), one of the two versions is adopted in Table 5.11 for each route for each relevant period (according to some pre-defined 'selection rule') to calculate the temporal averages (presented in Column E(P)). In the explanation of PPM (c.f. Network Rail, 2017), only a general 'selection rule' is adopted (without specific definitions): TAL = 5 mins for 'London and South East or regional services', and TAL = 10 mins for 'long distance services'. Here, for the convenience of calculation, a 'selection rule' (for choosing between the two versions of ΔRPM for each route for each period) based on the spatial distance between the origin and destination stations of each specific route is adopted: the version under TAL = 5 is adopted if the distance is less than 90 miles, and the version under TAL = 10 is adopted if the distance is larger than 90 miles. The rightmost column NOTE details the information about each route: the distances have been obtained by searching Google Maps, the values of which have been derived from the shortest paths within the road networks. Under this specific 'selection rule', an average gain of 7.2 % in RPM (c.f. the bottom cell in the table) can be expected for the nine studied critical routes by adopting the proposed algorithmic approach in Chapter 4.

From the above explanation of Table 5.11, we can see that a PPM-style summary tends to be heavily dependent on the pre-defined ‘selection rule’, which might introduce an unnecessary extra increase in subjectivity. An alternative and less subjective way to report a synthesised index (i.e. the 7.2% calculated from Table 5.11) is to firstly determine its lower and upper bounds and then report an interval bounded by the two calculated extremes. In the context of the nine studied routes in the previous subsections, the upper bound of the average gain in RPM (brought by modifying pre-trip information according to the proposed algorithmic approach) can be obtained by summarising all those ΔRPMs (c.f. Subsections 5.3.2 ~ 5.3.10) under TAL = 5 mins, and the lower bound can be obtained by summarising all those ΔRPMs (c.f. Subsections 5.3.2 ~ 5.3.10) under TAL = 10 mins. That is, instead of locking ourselves in an endless debate about which routes should be categorised into which group (i.e. TAL = 5 or TAL = 10), we can avoid the introduction of a subjective ‘selection rule’ by reporting an interval bounded by two definite limits.

Tables 5.12 and 5.13 below respectively summarise the route-specific results under the two ‘extreme cases’ (i.e. all TALs are set to 5 mins and all TALs are set to 10 mins). It can be seen from the two tables that the average gain (in RPM) for the nine studied routes lies between 5.0 % and 11.3%.

Table 5.12 Summary based on TAL = 5 mins (unit: %)

	P1	P2	P3	P4	P5	E(P)
BSB	13.2	12.1			9.7	11.7
ECB	17.2	15.3	16.0	30.6	12.6	18.3
LMD	28.3	13.4	31.3	18.8	11.8	20.7
ILM		2.0	5.2			3.6
RBH	4.6	0.0			0.0	1.5
KYS			12.2	20.3	3.4	12.0
HMN			26.8	23.6	14.1	21.5
KWN				6.3	0.6	3.5
SML				11.8	6.5	9.2
						11.3

Table 5.13 Summary based on TAL = 10 mins (unit: %)

	P1	P2	P3	P4	P5	E(P)
BSB	4.2	3.0			5.4	4.2
ECB	9.7	6.1	6.7	13.0	4.7	8.0
LMD	12.3	8.0	13.6	3.6	3.5	8.2
ILM		2.0	4.1			3.1
RBH	4.0	0.0			0.0	1.3
KYS			5.1	9.2	2.1	5.5
HMN			14.0	7.9	3.4	8.4
KWN				2.9	1.1	2.0
SML				5.9	2.9	4.4
						5.0

Generally, the obtained results (in previous subsections) make sense, for the modified pre-trip information should to some degree improve punctuality and reliability: as explained in Subsection 4.3.2, the mechanism of the modified pre-trip information (generated from the proposed algorithmic approach) is to add to each critical itinerary extra allowance (i.e. time supplement) to reduce the impact of delays/disruptions, based on the historical performance of each particular itinerary. However, two questions arise when confronted with those specific results: Why would some of the studied routes expect more significant improvement in RPM than the others, by adopting the modified pre-trip information? What do those zero values (c.f. Table 5.5) mean? In order to answer these questions, it would be helpful to have a closer look at the relevant sample data that have been adopted in the corresponding analyses.

Figure 5.7 below presents the descriptive statistics of the sample data about the route RBH (Rugeley Trent Valley – Birmingham New Street – Hereford) during Period 2 (25 January 2016 – 18 March 2016). It has been shown in Table 5.5 (c.f. Subsection 5.3.5) that the modified pre-trip information (generated from the proposed algorithm of PBPM) could not bring improvement in RPM for this route during Period 2. In Figure 5.7 below, two distributions are presented to describe the underlying sample data, corresponding to the distribution of arrival delays under the assumption of unmodified pre-trip information (a), and the distribution of arrival delays under the assumption of modified pre-trip information (b), respectively.

AD1(mins)	percentage
-6	4.49
-5	30.34
-4	25.84
-3	10.67
-2	7.87
-1	3.93
0	1.69
1	2.81
2	1.69
3	1.12
4	1.12
5	0.56
14	0.56
16	0.56
18	1.12
20	0.56
23	0.56
35	0.56
56	1.69
57	0.56
58	0.56
60	0.56
116	0.56
[Min, 5)	91.6
[Min, 10)	92.1
N = 178	
Min = -6, Max = 116	

(a)

AD2(mins)	percentage
-15	1.69
-14	1.69
-13	1.12
-12	1.12
-11	1.69
-10	5.62
-9	4.49
-8	3.37
-7	13.48
-6	11.24
-5	20.22
-4	11.80
-3	5.62
-2	2.81
-1	1.12
0	0.56
1	1.69
2	0.56
3	1.12
4	0.56
5	0.56
11	0.56
13	1.12
14	0.56
17	0.56
20	0.56
30	0.56
46	0.56
49	0.56
53	0.56
54	1.12
58	0.56
114	0.56
[Min, 5)	91.6
[Min, 10)	92.1
N = 178	
Min = -15, Max = 114	

(b)

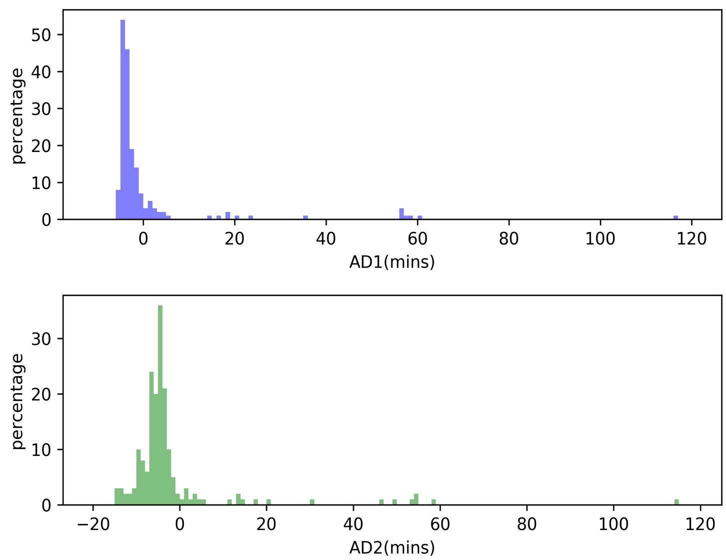


Figure 5.7 Distributions of arrival delays for RBH during Period 2
(NOTE: AD1 = $arr_a - arr_s$, representing the arrival delay calculated from the unmodified pre-trip information; AD2 = $arr_a - arr_m$, representing the arrival delay calculated from the modified pre-trip information)

It can be seen from Figure 5.7 that an apparent reason for the zero values under both scenarios (i.e. TAL = 5 and TAL = 10) is that the performance statistics are already quite good under unmodified pre-trip information. Specifically, an RPM_s (under TAL = 5) of 91.6% is already quite good for a transfer-involved rail route (c.f. Figure 5.7(a) and Table 5.5), and an RPM_s (under TAL = 10) of 92.1% is also among the best in the context of the nine studied routes. That is, the space left for improvement itself is limited in these two scenarios. However, good performance itself could not thoroughly explain these zero values. For example, the route ILM (Ilkley – Leeds – Middlesbrough) also has good performance under unmodified pre-trip information (c.f. Table 5.6) but improvements could still be expected. That is, there must be other factors exerting influence on the results. By further examining the two distributions in Figure 5.7, it is recognised that the modified pre-trip information (c.f. Figure 5.7(b)) does not truly change the distribution of arrival delays (c.f. Figure 5.7(a)). To help better understand the relevant issues, Figure 5.8 below is needed.

AD1(mins)	percentage
-2	5.66
-1	1.89
0	10.38
1	12.26
2	8.49
3	14.15
4	5.66
5	1.89
6	4.72
7	5.66
8	0.94
9	4.72
10	3.77
11	0.94
13	0.94
15	0.94
16	0.94
17	0.94
18	2.83
19	0.94
21	0.94
26	0.94
38	0.94
61	0.94
66	0.94
69	0.94
76	0.94
77	0.94
78	1.89
120	0.94
126	0.94
[Min, 5)	58.5
[Min, 10)	76.4
N = 106	
Min = -2, Max = 126	

(a)

AD2(mins)	percentage
-37	1.89
-36	0.94
-35	0.94
-33	0.94
-32	0.94
-31	0.94
-26	2.83
-25	1.89
-24	0.94
-23	2.83
-22	2.83
-21	3.77
-20	2.83
-19	8.49
-18	3.77
-17	1.89
-16	5.66
-15	1.89
-14	3.77
-12	1.89
-11	3.77
-10	2.83
-9	3.77
-8	3.77
-7	3.77
-6	2.83
-5	0.94
-4	1.89
-3	0.94
-2	2.83
0	0.94
1	0.94
2	1.89
3	2.83
4	0.94
6	0.94
9	0.94
12	0.94
17	0.94
24	0.94
40	0.94
44	0.94
47	0.94
57	1.89
58	0.94
64	0.94
101	0.94
103	0.94
[Min, 5)	86.8
[Min, 10)	88.7
N = 106	
Min = -37, Max = 103	

(b)

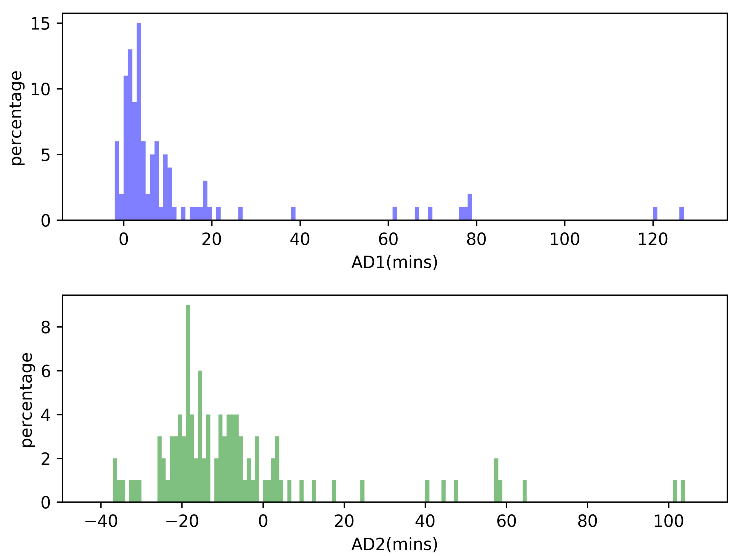


Figure 5.8 Distributions of arrival delays for LMD during Period 1
(NOTE: AD1 = $arr_a - arr_s$, representing the arrival delay calculated from the unmodified pre-trip information; AD2 = $arr_a - arr_m$, representing the arrival delay calculated from the modified pre-trip information)

Figure 5.8 above presents the descriptive statistics of the sample data about the route LMD (Liverpool Lime Street – Manchester Piccadilly – Doncaster) during Period 1 (12 October

2015 – 4 December 2015). Unlike the evaluation results (i.e. no improvements in RPM) for Figure 5.7, a significant improvement (in RPM) is observed for this route during this period (i.e. Period 1, c.f. Table 5.4 in Subsection 5.3.4).

To compare and identify the difference between Figure 5.7 and Figure 5.8, each of the four involved distributions (i.e. Figure 5.7(a), Figure 5.7(b), Figure 5.8(a), and Figure 5.8(b)) needs to be viewed as a combination of two parts, the cut-off point of which is the TAL (threshold for arrival lateness) adopted in the calculation of an RPM. For example, if we adopt a TAL of 5 minutes, then each of the four distributions in the two figures can be subdivided into two parts: those smaller-than-five observations (denoted by S5) and those larger-than-five observations (denoted by L5). Since each distribution has been sorted (by delay value) in ascending order, the S5 part corresponds to the upper end and the L5 part corresponds to the lower end.

By comparing Figure 5.7(a) with Figure 5.7(b), it can be seen that the modified pre-trip information (corresponding to Figure 5.7(b)) only changes the distribution of observations within each of the S5 group and the L5 group, but does not change the balance of power between S5 and L5. By contrast, if we compare Figure 5.8(a) with Figure 5.8(b), we can see that the modified pre-trip information (corresponding to Figure 5.8(b)) not only changes the distribution of observations within each of the S5 group and the L5 group, but also changes the balance of power between S5 and L5 (i.e. S5 goes up from 58.5% to 86.8%, c.f. Figure 5.8 and Table 5.4 in Subsection 5.3.4).

Based on the above investigations, it is recognised that the size of improvement (in RPM) the modified pre-trip information (generated from the proposed algorithmic approach) could bring depends, at least, on the following two factors. Firstly, it depends on the percentage of medium-sized (e.g. 5 ~ 30 mins) arrival delays that have occurred for a studied route during a given observation period. Secondly, it depends on whether the allowances added (by the proposed algorithmic approach) are sufficient to absorb those medium-sized arrival delays. Continue the examples of Figure 5.7 and Figure 5.8. It can be seen from Figure 5.7(a) that the percentage of medium-sized delays is relatively small (about 4% between 5 and 30), but in Figure 5.8(a) this percentage is relatively large (about 32% between 5 and 30). On the other hand, the allowances added for RBH during Period 2 (c.f. Figure 5.7(b)) are relatively small and could not change the balance of power between S5 and L5; in contrast, the allowances added for LMD during Period 1 (c.f. Figure 5.8(b)) are generally large and could 'push' some of the observations in L5 into S5.

In fact, it is the mechanism of the modified pre-trip information (generated by the proposed algorithmic approach) that results in that those added allowances are mainly used for coping with medium-sized delays. Recall that the specific calculation method adopted to generate modified pre-trip information has been detailed in previous sections (c.f. Subsection 4.3.2 and Subsection 4.3.8). Hence, a rough estimation of the general size of the allowances added (to a recommended list of critical itineraries) can be made with the aid of the real-world rail routes studied in this section. Suppose that the 'success rate' for a planned transfer is, on average, 80% for a given critical route (in the context of British railways) during a given observation period, and that the headway of the connecting leg is one hour (i.e. 60 mins, c.f. the nine studied routes in this section), then the allowance added to compensate for the risk of a missed transfer would be 12 minutes (i.e. 0.2×60) according to Step 5 of Algorithm 4 (c.f. Subsection 4.3.2). Meanwhile, if the average lateness of the connecting leg is +3 minutes for the same route during the same observation period, then an extra 3 minutes would be added to the allowance according to the algorithm proposed. That is, an average allowance of 15 minutes would be added to the unmodified pre-trip information about an 'ordinary' route described above, which would absorb those medium-sized arrival delays between 5 and 20 mins if a TAL of 5 minutes is adopted and would absorb those medium-sized arrival delays between 10 and 25 mins if a TAL of 10 minutes is adopted. Thereby, the size of improvement (in RPM) the modified pre-trip information (i.e. those added allowances) could bring depends, in the context of this illustrative example, on the percentage of arrival delays between 5 and 25 mins.

Until now, the answer to the two questions raised at the beginning of this subsection has been found out, with the aid of an explanation of the mechanism of the modified pre-trip information (generated by the proposed algorithmic approach). Looking back at those specific evaluation results of the nine studied routes, potential limitations of the proposed measure (i.e. RPM) itself have also been recognised in the investigation into the underlying sample data.

Figure 5.9 below provides such an illustrative example. This example is based on the evaluation results for the route ILM (Ilkley – Leeds – Middlesbrough) during Period 2 (25

January 2016 – 18 March 2016) and Period 3 (13 June 2016 – 5 August 2016). Here, the focus is no longer on same-period comparison between different versions of RPM, but is focused on same-indicator comparison between different observation periods. Specifically, the performance indicator adopted in Figure 5.9 is RPM_s under TAL = 5 (c.f. Table 5.6 in Subsection 5.3.6). Comparing the two values for the two periods (i.e. 88.9% and 84.5%), the adopted performance indicator tells us that this route performs better during Period 2 (corresponding to the value of 88.9%). However, when taking a closer look at the specific distributions of the sample data, we can find that although the percentage of 'successful realisations' is higher for Period 2, some key information is hidden about distribution of the sample data: the size of those 'failures' is also larger for Period 2. This inability to reflect the whole picture of the underlying sample data is a potential limitation in real-world applications of RPM and PPM (c.f. Network Rail, 2017), which indirectly explains why several auxiliary performance indicators such as CaSL (Cancellation and Significant Lateness) are also utilised by Britain's rail industry (c.f. Network Rail, 2017).

AD1 (mins)	percentage
-6	1.01
-5	1.01
-4	18.18
-3	16.16
-2	24.24
-1	10.10
0	7.07
1	4.04
2	5.05
3	2.02
5	2.02
10	1.01
11	2.02
14	1.01
21	1.01
53	1.01
56	1.01
60	1.01
118	1.01
[Min, 5)	88.9
N = 99	
Min = -6, Max = 118	

(a)

AD1 (mins)	percentage
-5	1.03
-4	8.25
-3	15.46
-2	28.87
-1	8.25
0	8.25
1	5.15
2	3.09
3	3.09
4	3.09
5	3.09
6	1.03
9	1.03
11	1.03
12	3.09
13	1.03
17	1.03
23	1.03
24	1.03
31	1.03
34	1.03
[Min, 5)	84.5
N = 97	
Min = -5, Max = 34	

(b)

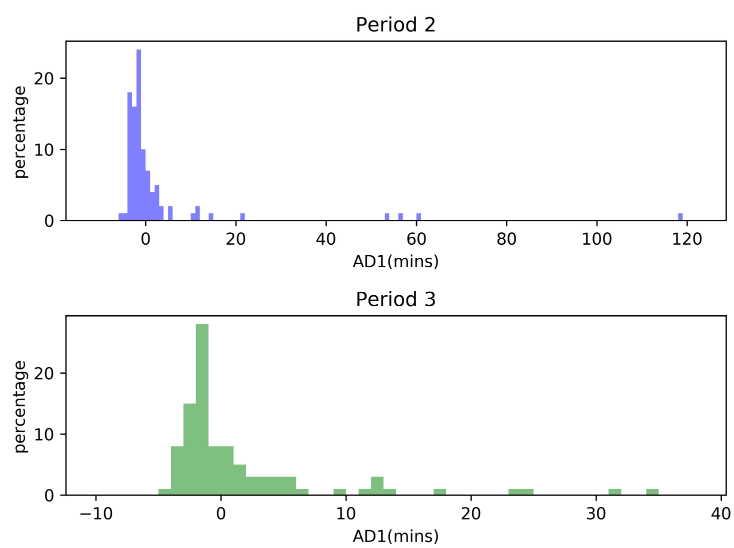


Figure 5.9 Distributions of arrival delays for ILM
[NOTE: AD1 = $arr_a - arr_s$, representing the arrival delay calculated from the unmodified pre-trip information; (a) corresponds to Period 2; (b) corresponds to Period 3]

5.3.12 Further analyses and Why would the proposed solution be better than the existing ones in tackling Critical Routes?

Although Tables 5.12 and 5.13 in the previous subsection have shown us the average gain (in RPM) that can be expected from applying PBPM to the nine studied (critical) routes, they cannot help learn more about the base case (i.e. unmodified pre-trip information based on the original schedules) and the treated case (i.e. modified pre-trip information based on PBPM). To address this gap, further analyses/evaluations have been conducted and Tables 5.14 and 5.15 below present the obtained results.

Table 5.14 Evaluation results based on RPM_s (TAL = 5 mins) (unit: %)

	P1	P2	P3	P4	P5	E(P)
BSB	72.1	74.2			75.3	73.9
ECB	72	75	71.1	54.9	83.7	71.3
LMD	58.5	75	55.9	66.1	81.2	67.3
ILM		88.9	84.5			86.7
RBH	79.2	91.6			93.7	88.2
KYS			73.7	58.2	84.2	72.0
HMN			55.3	57.9	72.9	62.0
KWN				80.6	89.9	85.3
SML				71.3	85.6	78.5
						76.1

NOTE: the meanings of the involved abbreviations/symbols can be found in Subsection 5.3.1 or Glossary

Table 5.15 Evaluation results based on RPM_p (TAL = 5 mins) (unit: %)

	P1	P2	P3	P4	P5	E(P)
BSB	85.3	86.4			84.9	85.5
ECB	89.2	90.3	87.1	85.5	96.3	89.7
LMD	86.8	88.4	87.3	84.8	92.9	88.0
ILM		90.9	89.7			90.3
RBH	83.8	91.6			93.7	89.7
KYS			85.9	78.4	87.7	84.0
HMN			82.1	81.5	87	83.5
KWN				86.9	90.4	88.7
SML				83.1	92.1	87.6
						87.4

NOTE: the meanings of the involved abbreviations/symbols can be found in Subsection 5.3.1 or Glossary

Both of the two tables adopt a TAL (threshold for arrival lateness) of 5 mins, corresponding to the relevant values presented in Table 5.12. The results in Table 5.14 are based on unmodified pre-trip information about scheduled arrival times, while the results in Table 5.15 are based on modified pre-trip information generated by PBPM. Comparing Tables 5.14 and

5.15, we can find that the 11.3% average gain (in RPM) that could have been obtained from applying PBPM to the nine studied routes (c.f. Table 5.12 in the previous subsection) actually corresponds to an increase from 76.1% (i.e. the average RPM in the base case) to 87.4% (i.e. the average RPM in the treated case).

Despite a few explanations made in Subsection 5.2.1, Subsection 5.2.2, and Subsection 5.3.11, the choice of delay thresholds (i.e. 5 and 10 minutes) throughout the evaluations in this section (i.e. Section 5.3) may still be questioned. After all, such a choice may have a direct influence on the obtained evaluation results. Admittedly, the adoption of the industry standards of British rail (i.e. 5 and 10 minutes) may still be classified as a (largely) subjective choice, for different railways in different countries may have different industry standards and even the industry standard for the same railway in the same country may itself change over time. The major consideration underlying the choice of delay thresholds in these RPM-based evaluations has been that adopting a consistent delay threshold (with the existing industry standard) would largely facilitate the comparison of the obtained route-level results with the existing network-level indices (e.g. PPM, c.f. Subsection 5.3.11 and Network Rail, 2017). Moreover, such a choice would to some degree facilitate international comparisons in future research as long as the relevant train operation records of railways outside the UK become legally accessible. Although different railways across the world adopt different delay thresholds (e.g. Dutch railways adopt 3 minutes as the industry standard), 5 and 10 minutes have a relatively large audience in European railways.

To help see the whole picture of the performances of PBPM under a series of different delay thresholds and meanwhile to some degree reduce the potential subjectivity in the choice of delay thresholds, a (quasi-) sensitivity analysis has been conducted and the obtained evaluation results are presented in Table 5.16.

Table 5.16 The evaluation results for RPM_s, RPM_p, and ΔRPM under different TALs

TAL (mins)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RPM_s (%)	50.9	59.9	67.3	72.7	76.1	78.9	81.2	82.8	84.0	85.3	86.5	87.5	88.4	89.2	89.8
RPM_p (%)	82.1	84.0	85.3	86.7	87.4	88.3	89.0	89.6	90.0	90.3	90.9	91.3	91.7	92.0	92.2
ΔRPM (%)	31.2	24.2	18.0	13.9	11.3	9.5	7.8	6.8	6.0	5.0	4.4	3.8	3.3	2.8	2.4

NOTE: the meanings of the involved abbreviations/symbols can be found in Subsection 5.3.1 or Glossary

From Table 5.16 we can see that the two columns under 5 (mins) and 10 (mins) have exactly the same values as those average values in Tables 5.12, 5.13, 5.14, and 5.15. If a delay threshold no greater than 5 mins was adopted, a two-digit increase in RPM would be expected, corresponding to a significant improvement in punctuality and reliability of the studied routes. If a delay threshold no less than 10 mins was adopted, a relatively small increase in RPM would be expected, corresponding to a modest improvement in punctuality and reliability of the studied routes. Overall, the size of the expected improvement gradually diminishes as the delay threshold adopted gradually increases. An incremental change to the delay threshold within the interval [0, 15] would not result in unexpected fluctuations in the obtained results. And even the least improvement (corresponding to a 15-minute delay threshold) is above zero.

After the above analyses of the gain side, let us conduct an analysis of the loss side. Since the essence of PBPM is the local treatment of those critical transfer-involved journeys by adding a floating extra allowance to the advertised arrival time of each of them based on its performance in the recent past, an identified potential loss/price of applying PBPM to those critical routes is an extension of the estimated journey times of the corresponding routes and a concomitant reduction/loss of competitiveness (/attractiveness) of the relevant recommendations. To help learn about the size of such a potential loss of competitiveness (/attractiveness), the relevant statistics have been computed and are presented in Tables 5.17 and 5.18.

Table 5.17 A summary of the relevant attributes of the studied critical routes

Route	**Periods**	**Itineraries** (denoted by scheduled departure times)	**Journey time** (nominal;mins)	***N*/Period**
BSB	P1, P2, P5	09:55, 10:59, 11:59, 12:59, 13:59, 14:59, 15:59, 16:59, 17:59, 18:59	139	200
ECB	P1 – P5	09:38, 10:37, 11:37, 12:37, 13:37, 14:37, 15:37, 16:37, 17:37, 18:37	189	200
LMD	P1 – P5	09:22, 10:22, 11:22, 12:22, 13:22, 14:22	133	120
ILM	P2, P3	10:10, 11:10, 12:10, 13:10, 14:10	122	100
RBH	P1, P2, P5	08:41, 09:43, 10:41, 11:41, 12:41, 13:41, 14:41, 15:41, 16:41	158	180
KYS	P3 – P5	08:30, 09:30, 10:30, 11:30, 12:30, 13:30, 14:30, 15:30	179	160
HMN	P3 – P5	08:28, 09:28, 10:28, 11:28, 12:28, 13:28, 14:28, 15:28, 16:28	82	180
KWN	P4, P5	08:53, 09:53, 10:53, 11:53, 12:53, 13:53, 14:53, 15:53, 16:53	127	180
SML	P4, P5	09:33, 10:26, 11:26, 12:26, 13:26, 14:26, 15:26	79	140
			134.3	

NOTE: the meanings of the involved abbreviations/symbols can be found in Subsection 5.3.1 or Glossary

Table 5.18 Average increase of advertised journey time (unit: mins)

	P1	P2	P3	P4	P5	E(P)
BSB	10.8	16.2			12.8	13.3
ECB	11.4	8.4	6.9	11.3	7.8	9.2
LMD	18.4	7.9	9.2	7.2	6.3	9.8
ILM		2.3	3.8			3.0
RBH	6.8	1.7			1.2	3.2
KYS			12.5	11.6	7.1	10.4
HMN			10.1	8.5	6.2	8.3
KWN				4.4	1.6	3.0
SML				8.3	4.4	6.3
						7.4

NOTE: the meanings of the involved abbreviations/symbols can be found in Subsection 5.3.1 or Glossary

It can be seen from Table 5.17 that the average nominal (/scheduled/advertised) journey time of these nine studied routes has been 134.3 mins. Meanwhile, we can learn from Table 5.18 that the allowance (contingency buffer) added by PBPM (for the nine studied routes for the five observation periods) has been on average 7.4 mins. That is, an average increase of 5.5% (7.4/134.3) in the nominal (/scheduled/advertised) journey time could have been expected. Such an increase in nominal (/scheduled/advertised) journey time would to some degree reduce the competitiveness (/attractiveness) of the relevant recommendations, which can be viewed as the 'price' of the proposed reliability (/robustness/punctuality) enhancing strategy (i.e. PBPM and PBPM+). However, although the proposed solution may not be the perfect to deal with those critical routes, it would at least be a (much) better solution than the existing ones in tackling critical routes. Why? Reasons are as follows.

Recall that the existing solutions/ideas have been generally categorised into two broad categories in Subsection 4.2.8 – CF (Competitiveness-First) ones and RF (Reliability-First) ones. Compared with those CF ones, the proposed solution could bring a noticeable improvement in reliability and punctuality (c.f. Tables 5.14 – 5.16 in this subsection) and meanwhile roughly/approximately maintain the competitiveness (/attractiveness) of the recommended itineraries (c.f. Tables 5.17 and 5.18 in this subsection), which means that the proposed solution may to some degree help increase the *customer stickiness*[17] of the relevant routes (c.f. Subsection 4.2.8 for explanations). Compared with those RF ones, the proposed solution could avoid significant reductions in competitiveness (/attractiveness) resulting from applying the existing RF solutions/ideas to critical routes (c.f. Subsection 4.2.8 for

[17] http://kwhs.wharton.upenn.edu/term/customer-stickiness/

illustrations), and hence the obtained estimations of reliability improvements (c.f. Tables 5.14 – 5.16 in this subsection) would be much more realistic and realisable than those derived from the existing RF solutions/ideas (in fact, the speculated improvements in reliability/robustness would be a rubber cheque in the case of critical routes; c.f. Subsection 4.2.8 for a detailed explanation of this issue).

5.4 Using Route-oriented Utility Measure (RUM) to quantify the effect of modified pre-trip information

5.4.1 Central idea

In the previous section, a route-level measure called RPM (Route-oriented Performance Measure) has been proposed that is able to evaluate the performance of a given transfer-involved (critical) route in terms of punctuality and reliability during a given observation period, and is easy to be extended to quantify the effect of modified pre-trip information. Although generally straightforward and easy to implement, the measure of RPM and the RPM-based analytical method have their limitations. Firstly, RPM is, in essence, a train-oriented performance indicator (rather than a passenger-oriented measurement). Recall that the concept of RPM is built upon an assumption of a representative passenger and an assumption of the existence of an absolute standard (i.e. a chosen threshold for arrival lateness (TAL, e.g. 5 mins and 10 mins) for determining whether a representative passenger is delayed. In reality, however, these underlying assumptions do not hold in many cases: a passenger inside a punctual train is still delayed if he/she has missed the previous connection; a passenger trip can still be punctual when taking a delayed train (Landex, 2008; Martin, 2014). That is, RPM (and RPM-based analyses) does not take into account the heterogeneity among the relevant passengers. Secondly, in a broader sense, the RPM-based analytical method is focused only on a single criterion of punctuality and reliability, and does not take into account other influencing factors on mode/itinerary choice such as the concomitant increase in advertised journey time (with a pre-modified arrival time). That is, although adding allowances to those critical itineraries could generally improve punctuality and reliability (c.f. the empirical results presented in the previous section), the overuse of allowances (resulting from uncontrollable errors within those performance statistics themselves) is likely to reduce the attractiveness and competitiveness of rail transport for a given route.

Based on the above considerations, a utility-based measure (analytical method) named RUM (Route-oriented Utility Measure) is proposed in this section to try to capture more realistic factors. Roughly speaking, RUM does not adopt an absolute standard (e.g. TAL in the calculation of RPM) for performance evaluation, but is a relative measure of how much difference a piece of modified pre-trip information could bring to a given transfer-involved (critical) route in terms of the overall utility of the relevant passengers. Technically, the RUM-based analytical method (to be presented later in this section) not only takes into account the inconvenience that may be caused by medium- to large-sized delays (as RPM does), but also takes into account the inconvenience that may be caused by small delays (e.g. those between 0 and 5 minutes) and early arrivals (i.e. those less than 0 mins). Before going to the specific technicalities, it should be noted that the RUM-based analytical method is largely experimental and is more like a *thought experiment* (compared with the RPM-based method) that is based on several 'bold' assumptions. However, this method could be employed as a convenient tool for quantifying the effect of modified pre-trip information when detailed data about train movements are available but detailed data about passenger activities are not available. Or at the very least, it could be a reference point for those interested researchers to refine the relevant theories.

5.4.2 Major assumptions

Assumption 5.1 (i.e. Assumption 5.1 in Subsection 5.2.2) Each of those identified critical routes is 'active': a given critical route (recommended by a journey planning system) would be utilised daily by a number of passengers; and even if the number is not large, it is greater than zero.

Assumption 5.4 (i.e. Assumption 5.4 in Subsection 5.2.2) The advertised arrival time of a given recommended itinerary is not necessarily equal to the scheduled arrival time in the timetable: it could be pre-modified by adopting, for example, the algorithmic approaches proposed in Chapter 4.

Assumption 5.5 Each recommended itinerary (for a given critical route) is treated as equally important in the computation of a specific RUM.

Assumption 5.6 There have been n (n > 0) passengers choosing a given studied itinerary.

Assumption 5.7 Each passenger choosing a particular itinerary (of a studied route) is associated with a *latest-tolerable arrival time* (LAT), which derives from his/her preferred arrival time (PAT) and the constraint of subsequent activities. And the (financial/reputational/psychological/physical) disutility beyond LAT is greater than the inconvenience caused by rescheduling (well in advance) the relevant activities.

Assumption 5.8 A passenger would minimise his/her expected disutility (i.e. maximise expected utility) when choosing among a list of recommended itineraries.

Assumption 5.9 There always exist a small group of 'unwary' passengers whose LATs are 'at the margin' (i.e. very close to the scheduled arrival time of a chosen itinerary).

The proposed measure of RUM (defined in the next subsection) is mainly built on the above six assumptions. Assumptions 5.1, 5.4, and 5.5 above are also the underlying assumptions of RPM (c.f. Subsection 5.2.2), in which Assumptions 5.1 and 5.4 are just a copy from Subsection 5.2.2 and Assumption 5.5 is a slightly modified version of Assumption 5.2 in Subsection 5.2.2. That is, four additional assumptions (i.e. Assumptions 5.6 – 5.9) are involved in the calculation of RUM.

Assumption 5.1 is the most basic assumption and is employed to emphasise that any evaluation or comparison would become meaningless if there exists no transport demand between a given pair of origin and destination stations. Assumption 5.5 (i.e. Assumption 5.2 in Subsection 5.2.2) is to emphasise that equal weights should be assigned to the recommended itineraries (for a given critical route) unless sufficient knowledge about the exact distribution of passenger flows is obtained, which can be viewed as an application of the *principle of indifference* (POI, c.f. Section 2.6). And similar to the role Assumption 5.4 plays in the calculation of RPM, this assumption is employed here to enable the comparison between the modified and unmodified pre-trip information to quantify the effect of modified pre-trip information.

Assumption 5.6 can be viewed as a relaxed version of Assumption 5.3 in Subsection 5.2.2. Recall that Assumption 5.3 plays an important part in the calculation of RPM: an 'average' or 'typical' passenger needs to be introduced to serve as the standard/reference point for evaluation. An implicit assumption is actually included in the description of Assumption 5.3: the number of passengers who have adopted a given recommended itinerary is nonzero (i.e. a positive integer). This is just what Assumption 5.6 says, but Assumption 5.6 does not further require the existence of an 'average' passenger.

Assumption 5.7 plays a key role in the calculation of RUM (to be presented later in the next subsection). It seems to be a bold assumption due to the fact that some of the relevant issues (i.e. LAT-related issues) have not been touched upon in previous transport studies. Despite disregarded in the literature, this does not mean that the relevant issues are unimportant. The concepts of PAT (c.f. Bates et al., 2001; Noland and Polak, 2002) and LAT (c.f. Senbil and Kitamura, 2004) have been mentioned mainly in the context of (macroscopic) economic studies with a focus implicitly placed on direct routes, whereas the focus of this section is placed on individual passengers (i.e. taking into account the heterogeneity among passengers on a microscopic level) and on transfer-involved rail routes. If an individual passenger's standpoint is adopted, there would be a diverse set of possible scenarios for a long-distance, transfer-involved rail journey (e.g. those identified critical routes in British railways), which renders a simple demarcation of journey purpose (i.e. commuting/leisure/business, c.f. Table 1.1 in Chapter 1) inappropriate. Here, in Assumption 5.7, the emphasis is placed on the possible existence of time-critical scenarios in inter-city rail travel. For example, a young man going from one city to another to attend a job interview, a journey to a major airport located in another city, a journey to watching a sports game in another city, etc. Although these scenarios may be regarded as 'untypical', they should not be ruled out in an analytical model as long as there is not sufficient evidence to refute these possibilities.

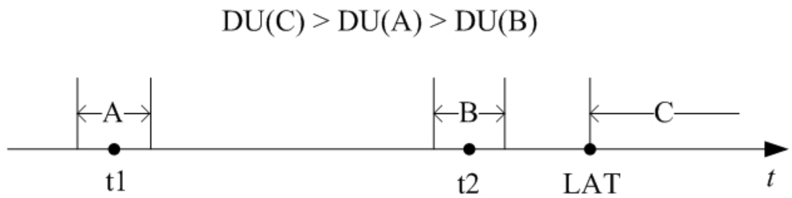


Figure 5.10 An illustrative example of Assumption 5.7 and Assumption 5.8
(NOTE: LAT = the latest-tolerable arrival time of a given passenger; DU = disutility; t2 = the scheduled arrival time of the chosen itinerary; t1 = the scheduled arrival time of the previous itinerary in the recommended list; A, B, C = the identifiers for the corresponding intervals)

Assumption 5.8 is a supplement to Assumption 5.7, which can be regarded as an application of the relevant concepts in Decision Theory (c.f. Section 2.6). Figure 5.10 (see above) gives a more tangible illustration of Assumptions 5.7 and 5.8: if a passenger arrived at the right hand side of his/her LAT (i.e. within Interval C), the disutility caused would be higher than if he/she had arrived within Interval A or Interval B; meanwhile, arriving within Interval B would cause the least disutility (compared with Interval A or C) based on Assumption 5.8 that a passenger would minimise his/her (expected) disutility (i.e. maximise (expected) utility, c.f. Subsection 2.6.2) when making a choice. Note that Figure 5.10 is only one of the possibilities (of rescheduling) to avoid DU(C): the figure is employed only for the convenience of illustration. That is, a passenger can always have a set of alternative options to avoid DU(C): 1) shift to another (intra-modal) recommended itinerary with a higher expected utility (e.g. Figure 5.10); 2) shift to another mode of transport and itinerary with a higher expected utility; 3) reschedule, well ahead of time, the subsequent activities at the destination; and 4) cancel the whole journey. In a word, DU(C) > DU(A) in the figure is not compulsory, and this further assumption underlying Figure 5.10 is only employed to make the relevant concepts more tangible. Moreover, it would be helpful for better understanding these two assumptions by comparing them with the relevant assumptions underlying previous studies such as Small (1982), Mahmassani and Chang (1986), and Bates et al. (2001). Firstly, Intervals A, B, and C in Figure 5.10 can be viewed as an extension of the concept of ‘indifference band’ in Mahmassani and Chang (1986), which are no longer narrowly defined in the context of urban car commuters. Secondly, in line with Small (1982) and Bates et al. (2001), Figure 5.10 also implicitly assumes the existence of ‘schedule disutility’. However, Figure 5.10 here does not make further assumptions on linearity in schedule disutility and on linearity in the overall disutility (i.e. a linear combination of journey time disutility, fare disutility, schedule disutility, etc) and only slightly involves ordinal (partially cardinal) utilities (i.e. DU(C) > DU(B) and DU(A) > DU(B) in Figure 5.10), which avoids extra unnecessary assumptions (according to Occam's razor) and avoids the potential problem of interpersonal utility comparisons (to be further explained in Section 5.5). Thirdly, the introduction of Assumption 5.7 is based mainly on the following two observations in reality. On the one hand, every person has exactly 24 hours per day and 7 days per week: excluding those daily routines such as sleeping, eating, and working, the available time for trips is inherently limited and hence the existence of LATs is natural. On the other hand, an implicit assumption that schedule disutility can play a predominant role in affecting itinerary choice is based on the observation that the other factors such as fare and journey time are often the same (or quite similar) in British railways in most cases, for a given direct route (or each part of a given transfer-involved route) is in most cases operated by a single rail operator in Britain's passenger rail system.

Assumption 5.9 is employed to reflect the heterogeneity in passengers' perception of potential delays. It should be noted that these 'unwary' passengers may be very wary in daily life, but becomes 'unwary' when making an itinerary choice due to various reasons. For example, an overoptimistic estimation of the reliability of a recommended itinerary due to a lack of experiential information. Figure 5.11 below provides an illustrative example of Assumption 5.9.

Figure 5.11 An illustration of Assumption 5.9
(Suppose the scheduled arrival time of the chosen itinerary is 16:00, a passenger with an LAT of 16:02 or 16:04 is said to be 'unwary')

5.4.3 The analytical model

Based on the assumptions presented in the previous subsection, RUM (Route-oriented Utility Measure) can be defined and calculated by the following analytical model (i.e. Eq. (3) and Eq. (4)), which can be easily extended to quantify the effect of modified pre-trip information.

$$f(act_{r,p,j,k}) = \begin{cases} \frac{\Delta_1}{\Delta}, act_{r,p,j,k} \geq md_{r,p,j,k} \\ \frac{\Delta_2 - \Delta_3}{\Delta}, sch_{r,p,j,k} < act_{r,p,j,k} < md_{r,p,j,k} \\ -\frac{\Delta_1}{\Delta}, act_{r,p,j,k} \leq sch_{r,p,j,k} \end{cases}$$

$$\begin{aligned} &where: \\ &\quad \Delta_1 = md_{r,p,j,k} - sch_{r,p,j,k} \\ &\quad \Delta_2 = act_{r,p,j,k} - sch_{r,p,j,k} \\ &\quad \Delta_3 = md_{r,p,j,k} - act_{r,p,j,k} \end{aligned} \qquad \text{Eq. (3)}$$

$$RUM_{r,p} = \frac{\sum_j^J \sum_k^K f(act_{r,p,j,k})}{J \cdot K} \qquad \text{Eq. (4)}$$

In Equations (3) and (4) above, the meanings of the involved notations are listed in the following:

- Those subscripts r, p, j, k respectively represent route identifier, period identifier, date identifier, and itinerary identifier.
- The three symbols $sch_{r,p,j,k}$, $md_{r,p,j,k}$, and $act_{r,p,j,k}$ respectively correspond to the scheduled arrival time of a given itinerary, the pre-modified arrival time of the (same) itinerary, and the actual arrival time of the (same) itinerary. (c.f. arr_s, arr_m, and arr_a in Figure 5.1 in Subsection 5.2.2)
- The meanings of Δ_1, Δ_2, and Δ_3 have been explained in Eq. (3), and the meaning of Δ is to be explained later in a further explanation of the analytical model (in this subsection).
- The symbol $f(act_{r,p,j,k})$ is an evaluation function for measuring the percentage of passengers who could have gained in utility of a given (studied) itinerary, the specific form of which depends on the position of $act_{r,p,j,k}$ on the time axis relative to $sch_{r,p,j,k}$ and/or $md_{r,p,j,k}$.
- RUM is short for Route-oriented Utility Measure, which is the proposed utility measure to quantify the effect of modified pre-trip information.
- The two capital letters J and K respectively represent the number of days during a given observation period and the number of studied (critical) itineraries per day.

Although the involved notations have been briefly explained (see above), the mechanism of those equations has not been explained and how to determine several key parameters (e.g. $sch_{r,p,j,k}$, $md_{r,p,j,k}$, and $act_{r,p,j,k}$) remains unclear. In the following, a more detailed explanation is to be presented about the key parameters involved and the mechanism of the analytical model as a whole.

Firstly, it should be noted that the three involved parameters in Eq. (3) – $sch_{r,p,j,k}$, $md_{r,p,j,k}$, and $act_{r,p,j,k}$ – have the same meaning of the three symbols arr_s, arr_m, and arr_a in previous sections (c.f. Figure 5.1 in Subsection 5.2.2 and Eq. (2) in Subsection 4.3.8). The reason for the change of notations is mainly due to the consideration that the previous illustrations have been oriented to a given (studied) itinerary, whereas the emphasis here is placed on that each studied itinerary is a member of a given sample set. Since the meanings are the same, the methods adopted in the determination of arr_s, arr_m, and arr_a are also applicable to the determination of $sch_{r,p,j,k}$, $md_{r,p,j,k}$, and $act_{r,p,j,k}$. More specifically, the scheduled arrival times (denoted by $sch_{r,p,j,k}$ here) can be easily determined from the timetable data; the pre-modified (advertised) arrival times (denoted by $md_{r,p,j,k}$) can also be generated from the proposed algorithmic approach in Chapter 4 (c.f. Algorithm 4 in Section 4.3); and the (reconstructed) actual arrival times (denoted by $act_{r,p,j,k}$) can also be determined by adopting the method presented in Subsection 5.2.5. Here, special attention should be paid to the determination of $act_{r,p,j,k}$: despite different assumptions adopted in RPM-based analytical method (in the previous section) and RUM-based analytical method (in this section), the method proposed in Subsection 5.2.5 (i.e. Table 5.1) also applies to the determination of $act_{r,p,j,k}$ here in the RUM-based model. However, the interpretation of Table 5.1 (in Subsection 5.2.5) needs to be changed: it is no longer oriented to an 'average' passenger, but is oriented toward each individual passenger; and the reconstructed (actual) arrival time of a studied itinerary can be interpreted as the *most likely* arrival time for most of the relevant passengers (i.e. those who have chosen this itinerary).

Once the three involved parameters (i.e. $sch_{r,p,j,k}$, $md_{r,p,j,k}$, and $act_{r,p,j,k}$) are determined for each studied itinerary (during an adopted observation period), Eq. (3) can then be applied to conduct itinerary-level analysis. Since the three intermediate parameters – Δ_1, Δ_2, and Δ_3 – can be easily derived from $sch_{r,p,j,k}$, $md_{r,p,j,k}$, and $act_{r,p,j,k}$, the only remaining parameter to determine is Δ. In fact, Δ as the denominator is based on an implicit assumption made on the distribution of the LATs (latest-tolerable arrival times) of the relevant passengers (who have chosen a particular itinerary): since little is known about the distribution of their LATs, a uniform distribution is introduced (based on the principle of indifference, c.f. Section 2.6), bounded by the scheduled arrival time of the chosen itinerary and an unknown but finite upper bound.

Figure 5.12 below provides an illustration of how to determine Δ. Δ represents the length of the interval between the scheduled arrival time ($sch_{r,p,j,k}$ = 16:00) of this itinerary and a finite upper bound denoted by LAT_max. Adopting the scheduled arrival time (16:00 in this illustrative example) as the lower bound of the distribution of LATs is mainly based on Assumption 5.9 in the previous subsection: the existence of 'unwary' passengers should not be ruled out unless there is sufficient evidence to refute this assumption. The upper bound LAT_max is unknown but should be finite: in reality, a passenger's daily activities would unavoidably be constrained by basic physical and psychological needs. In realistic applications (of Eq. (3)), scenario-based values could be assigned to LAT_max to facilitate the specific calculations. As soon as the lower and upper bounds of the distribution of LATs are determined, the parameter Δ (i.e. the length of the interval/domain) can be determined and hence be used to solve Eq. (3).

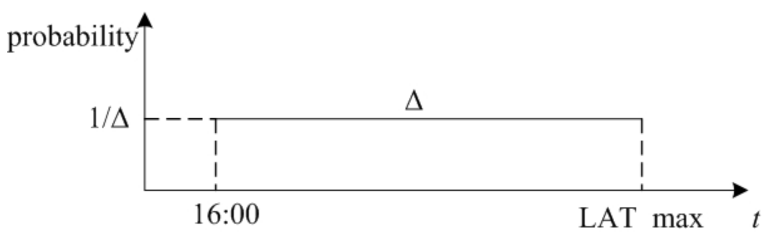


Figure 5.12 An illustration of the distribution of LATs of the passengers having chosen a studied itinerary (NOTE: the scheduled arrival time of this particular itinerary is 16:00; Δ is the length of the interval between 16:00 and LAT_max, which could be e.g. 1h, 2h, or even 3h, considering there might be some passengers having a considerably flexible schedule.)

Based on the introduction to the relevant parameters (in the above), each of those fractions (i.e. Δ_1/Δ, $(\Delta_2 - \Delta_3)/\Delta$ and $-\Delta_1/\Delta$) in Eq. (3) can then be interpreted as follows:

- If the actual arrival time is no less than the pre-modified arrival time (i.e. $act_{r,p,j,k} \geq md_{r,p,j,k}$), then those passengers whose LATs lying between the scheduled arrival time and the pre-modified arrival time would be better off if the pre-modified version has been adopted, either by shifting to the previous itinerary or by rescheduling the subsequent activities.

- If the actual arrival time lies between the scheduled arrival time and the pre-modified arrival time (i.e. $sch_{r,p,j,k} < act_{r,p,j,k} < md_{r,p,j,k}$), then those whose LATs lying between the scheduled arrival time and the actual arrival time would be better off, but those whose LATs lying between the actual arrival time and the pre-modified arrival time would be worse off.
- If the actual arrival time is no larger than the scheduled arrival time (i.e. $act_{r,p,j,k} \leq sch_{r,p,j,k}$), then those whose LATs lying between the scheduled arrival time and the pre-modified arrival time would be worse off.

Once the utility change in each studied itinerary is calculated, Eq. (2) can then be used to synthesise the results of all itineraries for a studied route during an adopted observation period. Here, equal weights are assigned to all involved itineraries based on the principle of indifference (c.f. Section 2.6 and Assumption 5.5 in Subsection 5.4.2).

5.4.4 A small numerical example

In order to help better understand the mechanism of the RUM-based analytical model (presented in the previous subsection), a small numerical example is employed in this subsection to illustrate the specific calculations.

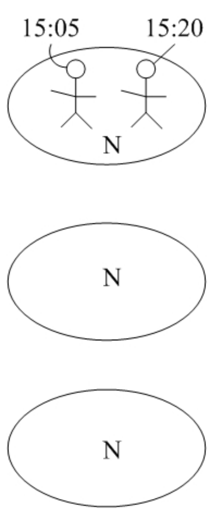
Table 5.19 below depicts an imaginary route containing three studied itineraries (denoted by i1, i2, and i3) per day and the observation period adopted is a particular day. The three parameters arr_s, arr_m, and arr_a (i.e. $sch_{r,p,j,k}$, $md_{r,p,j,k}$, and $act_{r,p,j,k}$) have all been determined for each of the three studied itineraries and are listed in Table 5.19. From the table (Table 5.19) we can see that the three studied itineraries respectively correspond to the three scenarios in Eq. (3) in the previous subsection.

Table 5.19 A fictitious critical route containing three studied itineraries

I	arr_s	arr_m	arr_a
i1	15:00	15:12	15:15
i2	16:00	16:15	16:09
i3	17:00	17:05	16:58

Figure 5.13 below adopts the values of arr_s, arr_m, and arr_a in Table 5.19 to illustrate how to calculate the RUM for this example route. To help better understand some abstractions, a *physical interpretation* is adopted here in this figure. Suppose each of the three itineraries in Table 5.19 corresponds to a set of N passengers (the three Ns in Figure 5.13 are treated as indistinguishable based on Assumption 5.5 in Subsection 5.4.2), and the LATs (latest-tolerable arrival times) of the N passengers are evenly distributed on the interval between two adjacent arrival times (e.g. [15:00, 16:00]) if the headway of this example route (i.e. 60 mins) is adopted as the parameter of Δ in Eq. (3). Then, we can apply Eq. (3) (presented in the previous subsection) to each of these three itineraries to calculate the itinerary-level utility change (i.e. $f(act_{r,p,j,k})$).

$$\Delta = H = 60$$
$$RUM = \overline{f(arr\_a)} = 1/18 \approx 5.6\%$$



$$f(arr\_a(i1)) = \frac{\Delta_1}{\Delta} = \frac{\frac{\Delta_1}{\Delta} \times N}{N} = \frac{\frac{12}{60} \times N}{N} = \frac{12}{60}$$

$$f(arr\_a(i2)) = \frac{\Delta_2 - \Delta_3}{\Delta} = \frac{\frac{\Delta_2}{\Delta} \times N - \frac{\Delta_3}{\Delta} \times N}{N} = \frac{\frac{9}{60} \times N - \frac{6}{60} \times N}{N} = \frac{3}{60}$$

$$f(arr\_a(i3)) = -\frac{\Delta_1}{\Delta} = \frac{-\frac{\Delta_1}{\Delta} \times N}{N} = \frac{-\frac{5}{60} \times N}{N} = -\frac{5}{60}$$

Figure 5.13 An illustration of how to calculate the RUM for the example in Table 5.19

For Itinerary 1 (i.e. i1 in Table 5.19), arr_a(i1) (= 15:15) is greater than arr_m(i1) (= 15:12). The evaluation function under this scenario is f(arr_a(i1)) = Δ_1/Δ = 12/60 (see Figure 5.13). What does this obtained result mean? It means that approximately 20 percent of the N passengers choosing this itinerary could have benefited if replacing arr_s(i1) (= 15:00) with arr_m(i1) (= 15:12). Alternatively, the result can be interpreted as the probability that a passenger choosing this itinerary could have gained in utility is approximately 0.2 if providing arr_m(i1) instead of arr_s(i1). Here, 'approximately' is used to emphasise that the result is only an estimation based on the available data and the relevant assumptions, and it is subject to uncontrollable errors from within the adopted data and assumptions themselves. Why could this (i.e. increase in utility) happen? This is because the information about arr_m(i1) (= 15:12) could alert those passengers lying between arr_s(i1) and arr_m(i1) (i.e. [15:00, 15:12]) to take actions (at an early stage) to avoid/reduce the disutility caused by being late (i.e. arriving later than their LATs), and the other passengers (i.e. those lying between [15:13, 16:00]) would be neither better off nor worse off. That is, those between [15:13, 15:15] would remain being late and those between [15:16, 16:00] would remain being on time, no matter whether arr_m(i1) (= 15:12) were informed.

Having obtained an understanding of the logic behind the analysis of Itinerary 1, the analyses of Itinerary 2 (i2) and Itinerary 3 (i3) can be understood in a similar way. Under the scenario of Itinerary 2 (i.e. arr_s < arr_a < arr_m), those between arr_s(i2) and arr_a(i2) (i.e. [16:00, 16:09]) would be better off while those between arr_a(i2) and arr_m(i2) (i.e. [16:09, 16:15]) would be worse off if the pre-trip information about arr_m(i2) (rather than that about arr_s(i2)) had been disseminated. Analogous to the analysis of Itinerary 1, the information about arr_m(i2) (= 16:15) could benefit those between arr_s(i2) and arr_a(i2) (i.e. [16:00, 16:09]) by enabling them to have sufficient time to take actions to avoid/reduce the disutility of being late. However, this piece of modified pre-trip information (about arr_m(i2)) would meanwhile increase the disutility of those between arr_a(i2) and arr_m(i2) (i.e. [16:09, 16:15]) by, for example, pushing them away from the most advantageous option to adopt a less advantageous option (e.g. shifting from the current itinerary (corresponding to Interval B in Figure 5.10 in Subsection 5.4.2) to the previous itinerary (corresponding to Interval A in Figure 5.10)). Here the key to understanding the increased disutility for those between arr_a(i2) and arr_m(i2) are Assumptions 5.6 and 5.8 in Subsection 5.4.2: since it is assumed that a number of passengers have adopted this itinerary (and rail transport) under the unmodified pre-trip information (Assumption 5.6), and that each passenger would minimise his/her expected disutility when making the itinerary choice (Assumption 5.8), a change of option (mode and itinerary) would to some degree increase the disutility of a passenger whose LAT lies between arr_a(i2) and arr_m(i2): although the passenger would be able to arrive at the destination station before his/her LAT by either adopting the most advantageous option (i.e. i2) or adopting a 'disutility-reduction' option (resulting from the pre-modified arrival time arr_m(i2)), the 'disutility-reduction' option would not be as advantageous as i2.

Based on the above considerations, the percentage of passengers who could have gained in utility is 3/60 for i2 (i.e. $f(arr\_a(i2)) = (\Delta_2 - \Delta_3)/\Delta = 9/60 – 6/60 = 3/60$) if the modified pre-trip information about arr_m(i2) (rather than that about arr_s(i2)) had been disseminated. The same logic applies to Itinerary 3: since arr_a(i3) (= 16:58) < arr_s(i3) (= 17:00), the pre-trip information about arr_m(i3) (= 17:05) would shift a passenger whose LAT is between 17:00 and 17:05 from the most advantageous option (i.e. i3) to a 'disutility-reduction' option, which would bring extra disutility to the passenger. Therefore, the percentage of passengers who could have gained in utility is -5/60 for i3 (i.e. $f(arr\_a(i3)) = -\Delta_1/\Delta = - 5/60$) if the unmodified pre-trip information about arr_s(i3) had been replaced with the modified pre-trip information about arr_m(i3).

Averaging the three obtained itinerary-specific indices, a route-level measure of RUM can then be calculated (i.e. 5.6% in Figure 5.13). That is, the modified pre-trip information (about pre-modified arrival times) could have enabled approximately 5.6% passengers to gain in utility for the studied route during the studied period. Note that here the example is fictitious, but those results obtained from the analyses of the identified critical routes in British railways (to be presented in the next section) are all based on large samples (containing hundreds of records) of real-world data.

5.5 Analyses of several identified critical routes using RUM

5.5.1 Data preparation

In this section, a number of identified critical routes in Britain's passenger rail system are to be analysed using RUM (Route-oriented Utility Measure) proposed in the previous section (i.e. Section 5.4). The aim of these analyses is twofold: on the one hand, they are utilised to quantify the effect of the modified pre-trip information (generated from the proposed algorithmic approach in Chapter 4) on the studied routes from a different perspective (with the RPM-based analytical method presented in Section 5.2); on the other hand, they are employed to enrich the understanding of these identified critical routes in British railways.

The data adopted to conduct these analyses are the same with those adopted in Section 5.3: historical train movements data about the relevant critical routes have been collected from Realtime Trains (RTT) during a 18-months period between September 2015 and March 2017.

The same list of nine studied routes (with that in Section 5.3) is adopted here, in which each route has been identified as critical during the 18-months period:

Bournemouth – Southampton Central – Brighton
Ebbw Vale Town – Cardiff Central – Birmingham New Street
Liverpool Lime Street – Manchester Piccadilly – Doncaster
Rugeley Trent Valley – Birmingham New Street – Hereford
Ilkley – Leeds – Middlesbrough
London Kings Cross – York – Scarborough
Harwich Town – Manningtree – Norwich
Knottingley – Wakefield Kirkgate – Nottingham
Sudbury (Suffolk) – Marks Tey – London Liverpool Street

Four observation periods (again, the same with those in Section 5.3) are selected to conduct the analyses of these nine routes, each of which contains 2-months (8-weeks) historical data (c.f. Subsection 5.2.4): Period 1 (P1) is between 12 October 2015 and 4 December 2015, Period 2 (P2) is between 25 January 2016 and 18 March 2016, Period 3 (P3) is from 13 June 2016 to 5 August 2016, Period 4 (P4) is from 3 October 2016 to 25 November 2016, and Period 5 (P5) is from 16 January 2017 to 10 March 2017.

The sampling method adopted in generating the modified pre-trip information is a semi-dynamic method based on an assumption of ‘the nearer, the more similar’ (c.f. Subsection 5.2.4). The sample size adopted for calculating RUMs is 4 weeks (c.f. Subsection 5.2.4).

Moreover, the specific technicalities used in reconstructing the actual arrival times can be found in Subsection 5.2.5.

5.5.2 The results

Based on the analytical model, the available data, and the relevant technicalities, the results from RUM-based analyses of the nine studied critical routes have been obtained and are presented in Table 5.20 below. The meanings of the involved notations are listed in the following:

- P1, P2, P3, P4, and P5 respectively correspond to Period 1 (12 October 2015 ~ 4 December 2015), Period 2 (25 January 2016 ~ 18 March 2016), Period 3 (13 June 2016 ~ 5 August 2016), Period 4 (3 October 2016 ~ 25 November 2016), and Period 5 (16 January 2017 ~ 10 March 2017).
- E(P) represents the average over the five observation periods (i.e. P1 ~ P5).
- N per P means the number of analysed itineraries during each of the five observation periods (i.e. P1 ~ P5).
- All the real numbers in Columns P1 ~ P5 and Column E(P) represent percentages (e.g. 0.61 means 0.61%).
- BSB represents the route Bournemouth – Southampton Central – Brighton.
- ECB represents the route Ebbw Vale Town – Cardiff Central – Birmingham New Street.
- LMD represents the route Liverpool Lime Street – Manchester Piccadilly – Doncaster.
- ILM represents the route Ilkley – Leeds – Middlesbrough.
- RBH represents the route Rugeley Trent Valley – Birmingham New Street – Hereford.
- KYS represents the route London Kings Cross – York – Scarborough.
- HMN represents the route Harwich Town – Manningtree – Norwich.
- KWN represents the route Knottingley – Wakefield Kirkgate – Nottingham.
- SML represents the route Sudbury (Suffolk) – Marks Tey – London Liverpool Street.
- The value (i.e. 2.8) in the bottom cell of Column E(P) is the average of the six positive values in the column.

Table 5.20 The evaluation results for the nine studied critical routes using RUM (unit: %)

	P1	P2	P3	P4	P5	E(P)	N per P
BSB	0.61	5.76			0.45	2.3	200
ECB	-3.63	-2.97	-3.07	-8.31	-4.58	-4.5	200
LMD	-11.19	3.47	-1.02	-2.51	2.41	-1.8	120
ILM		2.71	2.92			2.8	100
RBH	4.34	1.95			1.66	2.7	180
KYS			5.15	1.12	6.05	4.1	160
HMN			-2.53	-1.2	1.39	-0.8	180
KWN				2.31	2.03	2.2	180
SML				2.54	2.19	2.4	140
						2.8	

Note that the parameter Δ in the analytical model (c.f. Eq. (3) in Subsection 5.4.3) has been set to 60 minutes (i.e. the headway of these studied routes) in the analyses. Moreover, it should be noted that those blank cells in the above table (Table 5.20) are either due to planned/predictable cancellations (e.g. BSB, c.f. Subsections 4.3.7 and 5.3.2) or due to the changes in the list of identified critical routes (e.g. KWN did not enter the list during P1 – P3, c.f. Subsection 5.3.9).

When looking at those specific evaluation results in Table 5.20, the first reaction may be a shock: Why could several of these routes (i.e. ECB, LMD, and HMN) be associated with negative values? Why do those RPM-based counterparts of these negative results (c.f. Subsections 5.3.3, 5.3.4, and 5.3.8) reveal a totally different effect of the modified pre-trip information about these routes (i.e. ECB, LMD, and HMN)? In order to get a better understanding of these 'abnormal' results, an in-depth investigation into the sample data about and the characteristics of the relevant routes has been conducted, the findings of which are to be presented in the next subsection.

Moreover, the gains in RUM brought by the proposed algorithmic approach (corresponding to those positive decimals in Table 5.20) seem to be relatively small: What do these modest gains in RUM mean? Are they worth pursuing? Such questions are to be answered in the next subsection, with the aid of illustrative examples.

5.5.3 Interpretation

As mentioned in the previous subsection, some of the evaluation results in Table 5.20 seem to be 'abnormal', indicating that the effect of the modified pre-trip information (generated from the proposed algorithmic approach) is negative on the corresponding routes (i.e. ECB, LMD, and HMN) in terms of RUM (Route-oriented Utility Measure). More strangely still, the counterparts of these negative results in the RPM-based analyses (c.f. Subsections 5.3.3, 5.3.4, and 5.3.8) reveal that significant improvements could be expected in terms of RPM (Route-oriented Performance Measure).

After a comprehensive examination of the relevant data and the technicalities involved in the proposed analytical models, the following four aspects have been recognised as the most possible reasons for those 'abnormal' (negative) results in Table 5.20.

Firstly, those 'abnormal' results may be attributed to the difference between the mechanism of the RPM-based method and that of the RUM-based method. Recall that the RPM-based analytical method (c.f. Sections 5.2 and 5.3) is mainly built on an assumption of an 'average' passenger and an assumption of an absolute standard/threshold (e.g. 5 mins lateness) for determining whether an 'average' passenger has been delayed. That is, those small delays (e.g. < 5 mins) and early arrivals (i.e. negative values of delays) are regarded as 'successful realisations' in the context of RPM-based method. By contrast, under the analytical framework of RUM, those small delays and early arrivals would also be likely to increase the overall disutility associated with a studied route, considering the heterogeneity in passengers' perception of delays. Moreover, as illustrated in Subsection 5.3.12, those significant improvements in RPM the modified pre-trip information could bring to certain routes (e.g. LMD) can largely be attributed to a combination of a relatively high percentage of medium-sized delays and relatively generous allowances added (c.f. the empirical results presented in Subsection 5.3.12). In the RUM-based analytical model, however, a combination of medium-sized delays and generous allowances would be likely to introduce a lot of negative items (c.f. Eq. (3) in Subsection 5.4.3) and hence would be likely to lead to a decrease in the overall utility.

Secondly, those 'abnormal' results may have been caused by the inherent imperfections in the RUM-based model itself. As has been emphasised in Section 5.4, the RUM-based analytical model is largely experimental and is built on several ‘bold’ assumptions. More specifically, several applications of the principle of indifference (POI, c.f. Section 2.6) would be likely to lead to systematic errors in the evaluation results. For example, in the synthesis of itinerary-level indices in Eq. (4) (c.f. Subsection 5.4.3), equal weights are assigned to all involved itineraries (based on Assumption 5.5), which may lead to a biased result in the scenario that some of the involved itineraries correspond to significantly more passengers than the others during a given observation period.

Thirdly, the timetable design (at the tactical planning phase, c.f. Section 2.5) of the relevant routes may have resulted in those ‘abnormal’ values. Here, the two routes of ECB (Ebbw Vale Town – Cardiff Central – Birmingham New Street) and HMN (Harwich Town – Manningtree – Norwich) are employed to serve as illustrative examples (see Figures 5.14 ~ 5.17). Recall that in the introduction to the fundamentals of railway timetabling and operations (c.f. Section 2.5), two seemingly unrelated concepts have been respectively explained: *working timetable* and *running time supplement*. Roughly speaking, a working timetable is the counterpart of a published passenger timetable, which contains more technical details and is targeted at rail industry professionals. And running time supplements are added to the published passenger timetables for certain (direct) train lines to increase their robustness under small delays. Recall also that in the introduction to the concept of running time supplement (in Section 2.5) the operational practice in British railways has also been briefly mentioned: in Britain, running time supplements are not explicitly defined but are included in the timetables. After an in-depth investigation into the relevant published timetables and working timetables of the nine studied routes in this section, it is recognised that running time supplements have been implicitly included in the published timetables for several involved train lines. Figures 5.14 ~ 5.17 provide some illustrative examples of these implicitly added running time supplements, in which Figures 5.14 and 5.15 correspond to the feeder leg of ECB and Figures 5.16 and 5.17 correspond to the connecting leg of HMN.

www.realtimetrains.co.uk/train/P71682/2017/01/11/advanced

Mileage				WTT		GBTT		Realtime			Route		Allowances		
M	Ch	Location	Pl	Arr	Dep	Arr	Dep	Arr	Dep	Dly	Line	Path	Eng	Pth	Prf
0	0	Ebbw Vale Town [EBB] where this service forms from 2N07 from Bridgend			0938		0938		0938	RT					
1	32	Ebbw Vale Parkway [EBV]		0940½	0941	0941	0941		No report						
6	38	Llanhilleth [LTH]		0949	0949½	0949	0949		No report						
9	22	Newbridge [NBE]		0955	0955½	0955	0955		No report						
12	49	Cross Keys Jcn		pass	1001½				No report						
12	61	Crosskeys [CKY]		1002½	1003½	1003	1003		No report						
14	55	Risca & Pontymister [RCA]		1008	1008½	1008	1008		No report						
15	56	Risca South Jn		pass	1010½				No report						
16	6	Rogerstone [ROR]		1011½	1012½	1012	1012		No report						
17	46	Pye Corner [PYE]		1015½	1016	1016	1016		No report						
18	22	Park North Jcn		pass	1018				No report						
18	65	Park Jn		pass	1019½			pass	1018	1E					
19	47	Ebbw Jn		pass	1022½			pass	1021	1E	ML		2		
23	20	Marshfield		pass	1027			pass	1025	2E	DM				
25	10	Wentloog Frht Trmnal (Ews)		pass	1028½				No report						
27	21	Rumney River Bridge Jn		pass	1030				No report						
28	0	Pengam Jn		pass	1031				No report						
28	25	Moorland Road Junction		pass	1031				No report		RL				
28	75	Long Dyke Jn		pass	1033				No report		E				
29	58	Cardiff East		pass	1034½				No report						
29	70	Cardiff Central [CDF] where this service forms 5N13 to Cardiff Central	0	1035		1038		1041		6L					

Figure 5.14 Running time supplements Example One (Source: www.realtimetrains.co.uk, accessed 15 Jan 2017): the column under 'WTT' is the working timetable and 'GBTT' corresponds to the published timetable; Figures 5.15~5.17 below have the same format.

www.realtimetrains.co.uk/train/P71073/2017/02/20/advanced

Mileage				WTT		GBTT		Realtime			Route		Allowances		
M	Ch	Location	Pl	Arr	Dep	Arr	Dep	Arr	Dep	Dly	Line	Path	Eng	Pth	Prf
0	0	Ebbw Vale Town [EBB] where this service forms from 2N21 from Cardiff Central			1637		1637		1637	RT					
1	32	Ebbw Vale Parkway [EBV]		1639½	1640	1640	1640		No report						
6	38	Llanhilleth [LTH]		1648	1648½	1648	1648		No report						
9	22	Newbridge [NBE]		1654	1654½	1654	1654		No report						
12	49	Cross Keys Jcn		pass	1700½				No report						
12	61	Crosskeys [CKY]		1701½	1702½	1702	1702		No report						
14	55	Risca & Pontymister [RCA]		1707	1707½	1707	1707		No report						
15	56	Risca South Jn		pass	1709½				No report						
16	6	Rogerstone [ROR]		1710½	1711½	1711	1711		No report						
17	46	Pye Corner [PYE]		1714½	1715	1715	1715		No report						
18	22	Park North Jcn		pass	1717				No report						
18	65	Park Jn		pass	1718½			pass	1719	RT					
19	47	Ebbw Jn		pass	1721½			pass	1723¼	1L	ML		2		
23	20	Marshfield		pass	1726			pass	1727	1L	DM				
25	10	Wentloog Frht Trmnal (Ews)		pass	1727½				No report						
27	21	Rumney River Bridge Jn		pass	1729				No report						
28	0	Pengam Jn		pass	1730				No report						
28	25	Moorland Road Junction		pass	1730				No report		RL				
28	75	Long Dyke Jn		pass	1732				No report		E				
29	58	Cardiff East		pass	1733½				No report						
29	70	Cardiff Central [CDF] where this service forms 5B69 to Cardiff Central	4B	1734		1737		1738		4L					

Figure 5.15 Running time supplements Example Two (Source: www.realtimetrains.co.uk, accessed 21 Feb 2017)

From Figures 5.14 and 5.15, it can be seen that a 3-minutes time supplement is added to Cardiff Central (the transfer station) for this feeder line, which means that the scheduled window between the feeder line and the connecting line is implicitly increased by 3 minutes. A conceivable effect of these implicitly included time supplements is a reduced risk of missed transfers, and hence a reduced percentage of large arrival delays and reduced extra allowances added by the proposed algorithmic approach in Chapter 4, which might introduce a lot of negative items (c.f. Eq. (3) in Subsection 5.4.3) and hence would lead to a decrease in the overall utility.

www.realtimetrains.co.uk/train/G01274/2017/01/12/advanced

33	24	Chelmsford Brick House Xov		pass	0930½				No report			
35	74	Hatfield Peverel [HAP]	2	pass	0933			pass	0933¾	RT		
38	48	Witham [WTM]	3	pass	0935½			pass	0935¾	RT		
42	18	Kelvedon [KEL]	2	pass	0938			pass	0938	RT		
46	49	Marks Tey [MKT]	2	pass	0940½			pass	0941	RT		1
50	72	Colchester Signal Co4013		pass	0945				No report			
51	52	Colchester [COL]	2	0946	0948	0946	0947	0946	0948¾	RT		
59	35	Manningtree [MNG]	3	0955½	0956½	0955	0955	0956½	0958	1L		1
59	69	Manningtree North Jn		pass	0957½				No report			
67	67	Ipswich Halifax Junction		pass	1004½			pass	1005¾	1L	DL	
68	59	Ipswich [IPS]	3	1006½	1008½	1007	1008	1007¾	1010	1L		
69	41	Ipswich East Suffolk Jn		pass	1010			pass	1011¾	1L		
70	37	Ipswich Europa Junction		pass	1011				No report			
77	7	Needham Market [NMT]		pass	1015½				No report			
80	3	Stowmarket D.G.L.		pass	1017½				No report			
80	9	Stowmarket [SMK]	2	pass	1017½			pass	1020½	3L		
82	42	Haughley Jn		pass	1019½			pass	1022½	3L		
94	43	Diss [DIS]	1	1028	1029	1028	1029	1031	1032½	3L		2
113	31	Trowse Jn		pass	1045			pass	1046¾	1L		
113	68	Trowse Swing Bridge		pass	1046				No report		D	
114	11	Thorpe Junction		pass	1046½				No report		W	
114	40	Norwich [NRW]	1	1048		1050		1050		2L		

Figure 5.16 Running time supplements Example Three (Source: www.realtimetrains.co.uk, accessed 15 Jan 2017)

www.realtimetrains.co.uk/train/G01306/2017/02/20/advanced

33	24	Chelmsford Brick House Xov		pass	1230½				No report			
35	74	Hatfield Peverel [HAP]	2	pass	1233			pass	1235¾	2L		
38	48	Witham [WTM]	3	pass	1235½			pass	1237½	2L		
42	18	Kelvedon [KEL]	2	pass	1238			pass	1239¾	2L		
46	49	Marks Tey [MKT]	2	pass	1240½			pass	1242¼	1L		1
50	72	Colchester Signal Co4013		pass	1245				No report			
51	52	Colchester [COL]	2	1246	1248	1246	1247	1246½	1248¼	RT		
59	35	Manningtree [MNG]	3	1255½	1256½	1255	1255	1255¼	1256¼	RT		1
59	69	Manningtree North Jn		pass	1257½				No report			
67	67	Ipswich Halifax Junction		pass	1304½			pass	1304¼	RT	DL	
68	59	Ipswich [IPS]	3	1306½	1308½	1307	1308	1307	1309¼	RT		
69	41	Ipswich East Suffolk Jn		pass	1310			pass	1310¾	RT		
70	37	Ipswich Europa Junction		pass	1311				No report			
77	7	Needham Market [NMT]		pass	1315½				No report			
80	3	Stowmarket D.G.L.		pass	1317½				No report			
80	9	Stowmarket [SMK]	2	pass	1317½			pass	1318¾	1L		
82	42	Haughley Jn		pass	1319½			pass	1320½	1L		
94	43	Diss [DIS]	1	1328	1329	1328	1329	1328¾	1329¾	RT		2
113	31	Trowse Jn		pass	1345			pass	1343¾	1E		
113	68	Trowse Swing Bridge		pass	1346				No report		D	
114	11	Thorpe Junction		pass	1346½				No report		W	
114	40	Norwich [NRW]	1	1348		1350		1346¾		1E		

Figure 5.17 Running time supplements Example Four (Source: www.realtimetrains.co.uk, accessed 21 Feb 2017)

From Figures 5.16 and 5.17, it can be seen that discrepancies between the published timetable and the working timetable exist not only at the transfer station (i.e. Manningtree) but also at the destination station (i.e. Norwich). At Manningtree, the scheduled departure time (of the connecting leg of HMN) in the working timetable is 1.5 minutes later than its counterpart in the published passenger timetable, indicating that the scheduled window between the feeder leg and the connecting leg for this route has been implicitly increased by 1.5 minutes. Meanwhile, a 2-minutes time supplement has been implicitly added to the destination station (i.e. Norwich). It is conceivable that these implicitly included time supplements have reduced to some degree the overall magnitude of arrival delays at Norwich, and hence might introduce a lot of negative items (c.f. Eq. (3) in Subsection 5.4.3) leading to a decrease in the overall utility.

Lastly, but not least, other external factors may have led to those 'abnormal' results. As mentioned in Subsection 5.2.4, there are various external factors that may influence train movements. Although a lot of effort has been put into the analyses of the nine studied routes (either using RPM or using RUM): the choice of the five observation periods (i.e. P1 ~ P5) has carefully controlled several external factors such as public holidays and half-yearly changes in the long-term timetable, and the impact of planned/predictable cancellations has also been controlled in the sampling process, there may still exist some undetectable or uncontrollable factors that exert influence on the evaluation results.

In the above, potential factors resulting in those negative values have been systematically sorted out. Now let us shift our focus from those negative values to those positive values. At the end of the previous subsection, two relevant questions about those positive decimals in Table 5.20 have also been raised: What do these modest gains in RUM mean? Are they worth pursuing? In the following, the Author tries to answer these short but tricky questions by re-examining the relevant theories and their underlying assumptions.

Firstly, it should be emphasised that each positive/negative decimal in Table 5.20 represents the percentage of passengers who could expect a utility increase/decrease, rather than the percentage increase/decrease in the overall utility. For example, the decimal 2.8 in the bottom cell of Table 5.20 means that on average 2.8% of the passengers choosing the six routes that are associated with positive values (in Table 5.20) would expect a utility increase if the proposed algorithmic approach is adopted, and meanwhile the other 97.2% passengers would neither gain or lose in utility. A fundamental difference between 'the percentage of passengers who could expect a utility increase/decrease' and 'the percentage increase/decrease in the overall utility' lies in that the former does not involve *interpersonal utility comparisons* but the latter does. According to Briggs (2017), the expected utility theory itself is far from perfect and one of its potential limitations is the so-called *problem of interpersonal utility comparisons*: Mike's utilities are constituted by Mike's preferences; Cathy's utilities are constituted by Cathy's preferences; Mike's utility 10 is not necessarily equal to Cathy's utility 10. Although a number of potential solutions to this problem have been put forward in the literature such as the concepts of 'extended preferences' and 'extended utility functions' proposed by Harsanyi (1997) and Adler (2014) and several other theoretical frameworks in welfare economics (c.f. Adler and Fleurbaey, 2016), these concepts and theories remain immature and have not been widely accepted. Since the RUM-based method is built upon the expected utility theory, it also suffers from this limitation: the obtained results cannot precisely tell 'the percentage increase/decrease in the overall utility' but can instead tell 'the percentage of passengers who could expect a utility increase/decrease'. To better understand the subtleties and complexities, let us do the following thought experiment. Note that the involved cardinal utilities (as advocated by Ng (1997)) in the following experiment are merely employed for explanation, and the RUM model itself does not require cardinal utilities (c.f. Section 5.4).

Let us firstly make a bold assumption that there exists an *absolutely impartial* judge who has experienced all the pleasures and sufferings of a wide range of different groups of people so that he/she/it can precisely assign to each relevant passenger an *objective utility* measured by *standardised/normalised/universal utils* (the units of a person's utility is called *utils*; here the terminology *standardised utils* means they are interpersonally comparable). Now suppose there are 100 passengers choosing the six routes (with gains in RUM) during a given period of time (e.g. a week), each of which has the following utility function: DU(A) = -20, DU(B) = -10, and DU(C) = -30 (based on Figure 5.10 in Subsection 5.4.2; all measured by *standardised utils*). Based on the results in Table 5.20, we know that three (2.8 ≈ 3) out of the 100 passengers could obtain utility gains, while the other 97 would stay unaffected. That is, three out of the 100 could benefit from the modified pre-trip information by shifting from the worst-case outcome (i.e. Interval C in Figure 5.10) to a not-too-bad outcome (i.e. Interval A in Figure 5.10), while 97 out of the 100 would neither gain or lose in utility (i.e. staying in

Interval B or Interval C or the interval between B and C). To simplify the estimation, let us further assume that the average (dis-)utility of the 97 unaffected passengers is -20 (derived from a mixture of -30 ~ -10). Then, we can calculate the percentage increase in the overall utility by the following equation: $\Delta U = 3\times[(-20) - (-30)] / |[3\times(-30) + 97\times(-20)]| = 1.5\%$. That is, only 1.5% increase in the overall utility of the 100 passengers can be obtained if we assume that all of them have the (same) following utility function: DU(A) = -20, DU(B) = -10, and DU(C) = -30.

Now let us make a slight modification to the above numerical example by assuming the 97 still have the aforementioned utility function (i.e. DU(A) = -20, DU(B) = -10, and DU(C) = -30) but the three have the following: DU(A) = -20, DU(B) = -10, and DU(C) = -100. In this new context, although only three out of the 100 can gain in utility, the percentage increase in the overall utility becomes non-negligible: $\Delta U = 3\times[(-20) - (-100)] / |[3\times(-100) + 97\times(-20)]| = 10.7\%$. By comparing the above two numerical examples (thought experiments), we can see that although an average gain of 2.8% in RUM may be regarded as insignificant, this does not mean that the percentage increase in the overall utility would not be likely to be significant. Hence, the potential benefit of the 2.8% gain in RUM should neither be overestimated nor be underestimated.

If the Reader finds the above thought experiments too ridiculous, there is also a non-utilitarian argument for supporting the minority (i.e. the 2.8%): all too often, policy makers tend to ignore the 'minority' and favour the 'majority' either by relying on their own limited knowledge and experience or by telling 'each individual to imagine the probability of his being in various positions, rather than having him identify with the individuals who will actually occupy various positions' (quoted from Kamm (1998), and this phenomenon is called 'a veil of ignorance'). Of course, in the context of transport studies, we do not need to care too much about those serious life-and-death issues discussed in Kamm (1998). Are these 'modest' improvements worth pursuing? Well, the answer may be 'it depends'. It depends on how to define 'worthiness' ('man is an animal suspended in webs of significance he himself has spun'[18]). If defined from the perspective of cost efficiency, the answer may be 'Yes': all the infrastructure manager (information provider) needs to do is just import those source codes in the appendices of this thesis into a spare computer and make the computer spend

[18] A quote from Clifford Geertz.

several seconds per week to help some of the rail passengers arrive at their destinations on time (without discounting the others' interests). That is, whether to pursue these 'modest' improvements does not involve win-lose situations but is more like a choice between getting some bonus and getting nothing.

5.6 Exploiting RPM and RUM to do more

In the previous sections of this chapter, two novel route-level measures have been developed and applied to analyse several identified critical routes in British railways. In general, the results presented in previous sections (c.f. Sections 5.3 and 5.5) provide empirical evidence to demonstrate the effectiveness of the algorithmic approach proposed in Chapter 4. Moreover, based on these results, conclusions/findings can also be drawn for each specific route in these RPM- or RUM-based analyses.

Apart from these basic applications, RPM- and RUM-based analytical methods can also be utilised in more potential applications in railway timetabling and operations. Three readily conceivable applications are to be briefly described in this section, which act as a spur to further exploration and argumentation in future research.

The first conceivable application is to use RPM (Route-oriented Performance Measure, c.f. Section 5.2) to conduct more detailed assessments of rail operations. Existing performance measures widely adopted in the industry such as PPM (Public Performance Measure, c.f. Network Rail, 2017) are largely aggregate indices, which would hide a lot of information about local operations. If rail operators would like to know details about the performance of specific routes (lines), then RPM would be a potentially useful indicator.

The second conceivable application is to employ RUM (Route-oriented Utility Measure, c.f. Section 5.4) as an additional optimisation criterion to assess and compare a set of candidate timetables in the timetabling phase (c.f. Section 2.5). As explained in the relevant literature (e.g. Goverde, 2005; Vromans, 2005; and Andersson, 2014), several candidate timetables would often be firstly generated for assessment and comparison before one of them could be chosen as the published version (i.e. the optimal among the candidates in terms of some predefined criteria such as robustness, stability, etc). Since the RUM-based method is devised to compare two timetables that have very similar scheduled departure and arrival times (c.f. Sections 5.4 and 5.5), it can also be extended to conduct pair-wise comparisons among several candidate timetables to determine which version is the optimal in terms of the overall utility of the relevant passengers. It should be noted that both RUM and RPM (c.f. the previous paragraph) are not limited to the assessment of transfer-involved routes (as shown in the Sections 5.2~5.5), they can be readily exploited to evaluate the large set of direct routes (lines) within a given railway network.

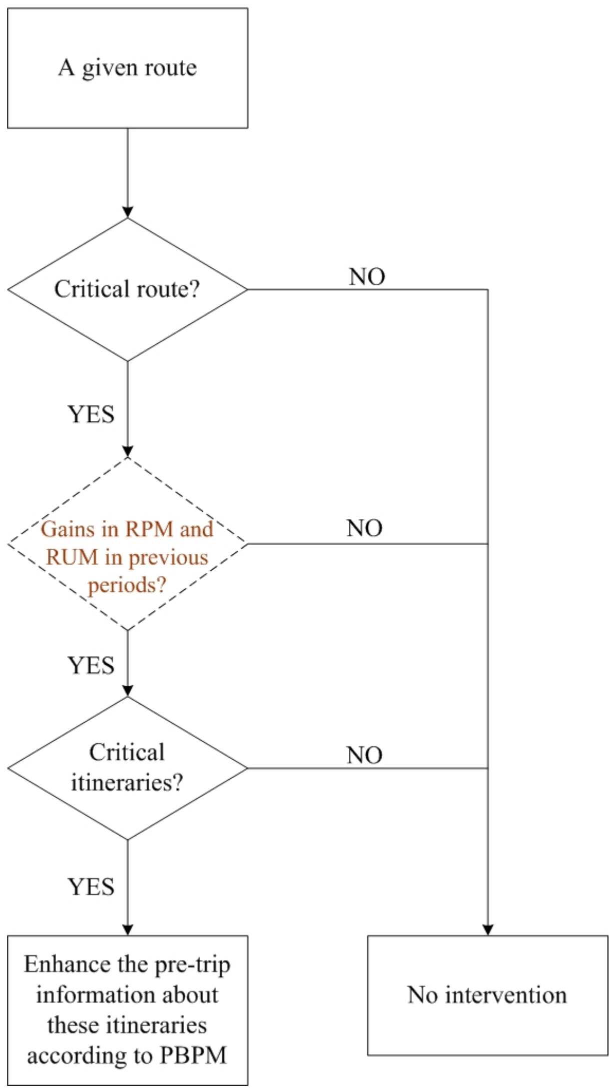


Figure 5.18 An augmented version of the algorithmic approach proposed in Chapter 4

The third conceivable application is to use RUM and RPM to augment the algorithmic approach proposed in Chapter 4. Figure 5.18 above provides an illustration of the augmented algorithm, which can be seen as a minor modification[19]of Figure 4.10 in Subsection 4.3.1. The major difference between this augmented version (Figure 5.18) and the original version (Figure 4.10) is that an additional conditional expression to check RUM and RPM has been introduced. To better understand the mechanism of this modified algorithmic approach (in Figure 5.18), let us look back at those technicalities and empirical results presented in the previous sections of this chapter.

Table 5.21 A summary of several key parameters of the nine studied routes in British railways

	SW	NTT	MTL	Supp1*	Supp2*	fType	stopNum
ECB	8.2*	2	9**	3	0	regional	7
HMN	5	1	5**	1~1.5	2	regional	4
LMD	11	3	8	0	0	long-distance	4
ILM	10	3	7	0	0	regional	5
KYS	8.1*	1	7.1	0	2	long-distance	3
KWN	6*	1	6**	1~1.5	0	regional	4
SML	4	1	3	0	0	regional	2
BSB	4.4*	1	3.4	0	0	long-distance	1
RBH	12	3	10**	1	3~6	regional	8

NOTE: SW = scheduled window; NTT = net transfer time; MTL = maximum tolerable lateness; Supp1 = the time supplement at the transfer station; Supp2 = the time supplement at the destination station; fType = the type of the feeder line; stopNum = the number of intermediate stops between the departure station and the transfer station; * = the corresponding value is a rough range or the average of different hours of a day; ** = the corresponding value has been adjusted by incorporating the corresponding time supplement; the meanings of the nine acronyms in the leftmost column can be found in Subsection 5.5.2.

Table 5.21 summarises a number of key parameters of the nine studied (critical) routes in British railways, which are assumed to be the potential factors that may exert influence on RPM and RUM. Comparing this table with those evaluation results presented in Table 5.12 and Table 5.13 (c.f. Subsection 5.3.11), we can firstly see that the reason why the proposed algorithmic approach (i.e. PBPM) would not bring significant gains (in terms of RPM) to the route RBH (Rugeley Trent Valley – Birmingham New Street – Hereford, c.f. Subsection 5.3.5) may be due to a combination of a relatively large MTL (Maximum Tolerable Lateness, c.f. Subsection 3.5.2) and a quite generous supplement/allowance at the destination station (i.e. Supp2 = 3~6): this combination would be likely to lead to a low percentage of medium-sized delays (due to the generous allowances that have been included in the timetable) and small extra allowances (generated from PBPM) added to the destination station (due to the

[19] Since PBPM (i.e. Algorithm 4: Performance-Based Pre-Modification) is the advocated, Algorithm 3 (Itinerary-oriented Performance Statistics) is omitted here.

low risk of missed connections resulting from a large MTL), which further elucidates the interpretations presented in Subsection 5.3.12.

With respect to those evaluation results in RUM-based analyses (c.f. Table 5.20 in Subsection 5.5.2), no clear patterns can be extracted from this table (i.e. Table 5.21): there are not significant differences between the three 'abnormal' routes (i.e. ECB, HMN, and LMD) corresponding to negative RUMs (listed at the upper rows of the table) and the other six in terms of these listed parameters. However, these values corroborate a previous observation that the distribution of train delays is influenced by many factors. Recall that it has been observed from the analyses of big data (c.f. Section 3.8) that the distribution of train delays (in British railways) is better modelled by those compound distributions (rather than those 'pure' distributions) such as q-exponential functions incorporating a number of different random variables (i.e. a number of different influencing factors). The specific parameters shown here in Table 5.21 (and those evaluation results about RPM and RUM presented in Sections 5.3 and 5.5) corroborate this finding: the impact of train delays on the nine studied transfer-involved routes cannot be simply explained by these listed parameters, implying that there must be other explanatory variables (i.e. influencing factors). Two conceivable additional factors are the percentage of unplanned cancellations and the level of crowdedness at stops en route. Since these two potential factors are difficult to measure without detailed relevant data, in-depth investigations of them are recommended for future research.

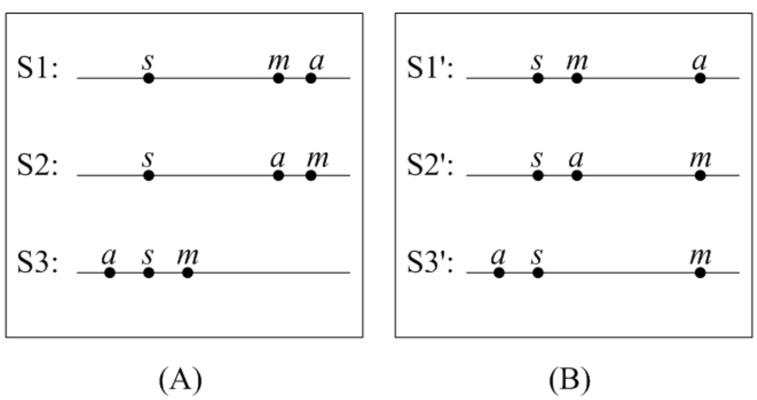


Figure 5.19 An illustration of several representative scenarios of the RUM model: (A) a collection of 'good-case' scenarios; (B) a collection of 'bad-case' scenarios. [NOTE: s = the scheduled arrival time, m = the pre-modified arrival time, and a = the actual arrival time]

Although it is difficult to exactly enumerate all the factors influencing the sign and magnitude of RUM, we can still identify some rough pattern by analysing the RUM model itself (i.e. Equations (3) and (4) presented in Subsection 5.4.3). Recall that in Equation (3) three classes of scenarios are differentiated by the relative positioning of the scheduled arrival time, the pre-modified arrival time, and the actual arrival time (of a given recommended itinerary), and that Equation (4) is just a synthesis of individual evaluations. To further investigate the RUM model, specific representative scenarios need to be firstly analysed. Figure 5.19 presents several fairly-good-case scenarios (illustrated in Collection A) and several fairly-bad-case scenarios (illustrated in Collection B).

In Figure 5.19 above, the three scenarios in Collection A (i.e. S1~S3) and the three in Collection B (i.e. S1'~S3') exactly correspond to the three classes of scenarios in Equation (3) (c.f. Subsection 5.4.3), respectively. And the relationship between the scenarios in Collection A and those in Collection B is one-on-one: S1 corresponds to S1', S2 corresponds to S2', and S3 corresponds to S3'. The only difference between the two collections lies in the size of $|ma|$ (i.e. the absolute difference of m and a): S1~S3 in Collection A have a significantly smaller $|ma|$ than their counterparts in Collection B (i.e. S1'~S3'), which could result in different signs (i.e. positive vs. negative) of the corresponding RUMs. Suppose there is a sample containing N=300 studied itineraries (of a studied route), among which exactly 100 itineraries belong to each of the three scenarios in Equation (3) (c.f. Subsection 5.4.3). Then, Collection A in Figure 5.19 can be derived by asking the following *what-if* question: what would the average scenario of each of the three classes in Equation (3) look like, if the calculated RUM (for this particular sample) turned out to be a large positive number? Similarly, Collection B can be derived by asking: what would the average scenario of each of the three classes in Equation (3) look like, if the calculated RUM (for this particular sample) turned out to be a large negative number? On the whole, Collection A guarantees that those positive items in Equation (3) have the upper hand, while Collection B enables those negative items in Equation (3) to predominate.

Although Collections A and B in Figure 5.19 can only be seen as special cases (i.e. each of the six scenarios in the figure could randomly occur in a sample), the six scenarios shown in the figure (i.e. S1~S3') are representative and can be further generalised: at least one of the two scenarios of S1 and S2 must predominate in a sample associated with a large positive RUM, while at least one of the two scenarios of S2' and S3' must predominate in a sample associated with a large negative RUM. That is, the sign and magnitude of RUM can generally reflect the degree of predictability of the actual arrival times of a studied route (by using the algorithmic approach proposed in Chapter 4): the pre-modified (advertised) arrival time (*m*) is close to the actual/recorded arrival time (*a*) in S1 and S2 (implying a high degree of predictability), whereas the pre-modified arrival time is significantly different from the actual arrival time in S2' and S3' (implying a low degree of predictability).

Based on the further analyses of the RUM model in the previous paragraphs, Figure 5.18 (i.e. the third conceivable application) can then be interpreted as follows. Rather than a one-off screening solely by executing CRF (Critical Routes Finder, c.f. Section 3.6), an additional second-round screening is introduced in the augmented framework (presented in Figure 5.18) to check if gains in RPM and RUM could be brought by the historical-data-based algorithmic approach (i.e. PBPM in Chapter 4) to an identified critical route (by CRF in the first round) in the previous observation periods (assuming detailed historical train movements data about the route are accessible). This additional screening step simultaneously takes into account the operator-oriented index (i.e. RPM) and the passenger-oriented index (i.e. RUM), and is mainly aimed at double-checking if there are other uncapturable factors (e.g. unquantifiable or undetectable factors) that may have a strong influence on the effectiveness of the proposed algorithmic approach (for a particular route): if there are not, then gains in RPM and RUM can be expected and hence the historical-data-based algorithmic approach can be readily adopted to improve the pre-trip information about this studied route; if there are, then losses in RUM can be expected and the historical-data-based approach cannot generate desirable predictions of arrival delays for this studied route, indicating either keeping the corresponding train schedules unchanged or devising other methods to improve the pre-trip information about this studied route. Note that since the proposed (historical-data-based) algorithmic approach would never result in losses in RPM (as illustrated and explained in Section 5.3), the sign and size of RUM becomes the decisive factor. In the specific context of the nine studied critical routes in British railways (c.f. Section 5.5), this means the proposed algorithmic approach may not be a good therapy for the three routes associated with negative RUMs (i.e. ECB, LMD, and HMN), indicating either no changes made to their original schedules or considering other approaches to dealing with them.

5.7 The limitation of the proposed measures and a potential solution

The limitation of the proposed measures in this chapter (i.e. RPM and RUM) lies mainly in several simplifying assumptions placed on passenger itinerary choice (e.g. resorting to an 'average' passenger to reconstruct/deduct the actual arrival time of a given itinerary), which can be further attributed to the unavailability of detailed data about passenger counts and passenger flows. That is, the quality of the obtained results from the proposed analytical methods in this chapter (i.e. the RPM-based method and the RUM-based method) is limited by the availability of detailed data about passenger counts and passenger flows. In fact, collecting passenger-related data has long been a challenging task in transport-related studies. However, this situation has been changing in recent years thanks to the development of the relevant devices.

A good example is the application of smart card data in a large number of relevant studies of urban public transport in the last decade or so (see e.g. Pelletier et al., 2011; Gordon et al., 2014). And van der Hurk (2015) even presents an application of smart card data in the context of Dutch railways. All of these previous studies could be adopted as a reference point for Britain's rail industry. Good news is a number of train operating companies (TOCs) in Britain have recently been rolling out smart card services: Figure 5.20 below provides an illustration of South West Trains, others having participated in this scheme include ScotRail, Southeastern, Southern, Thameslink and Great Northern, Greater Anglia, and c2c.

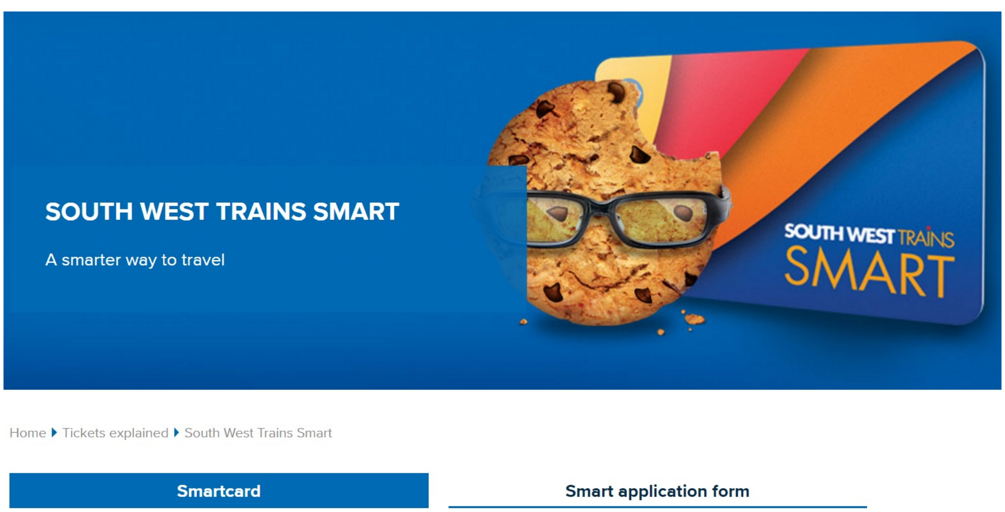


Figure 5.20 The Smartcard advertisements of South West Trains
(Source: www.southwesttrains.co.uk/tickets-explained/smartcard/, accessed 22 Feb 2017)

Despite the considerable potential underlying these newly adopted technologies, it may take some time to build an integrated and truly useful database that enables the relevant rail research. Observing the potential applications of the relevant technologies in the context of Britain's passenger rail system, here depicts a 'blueprint' of how to reconstruct/deduct passengers trajectories for those transfer-involved rail routes in future research (see Figure 5.21 below). The general idea is to make use of both the smart card data and those recorded by NRE (National Rail Enquiries).

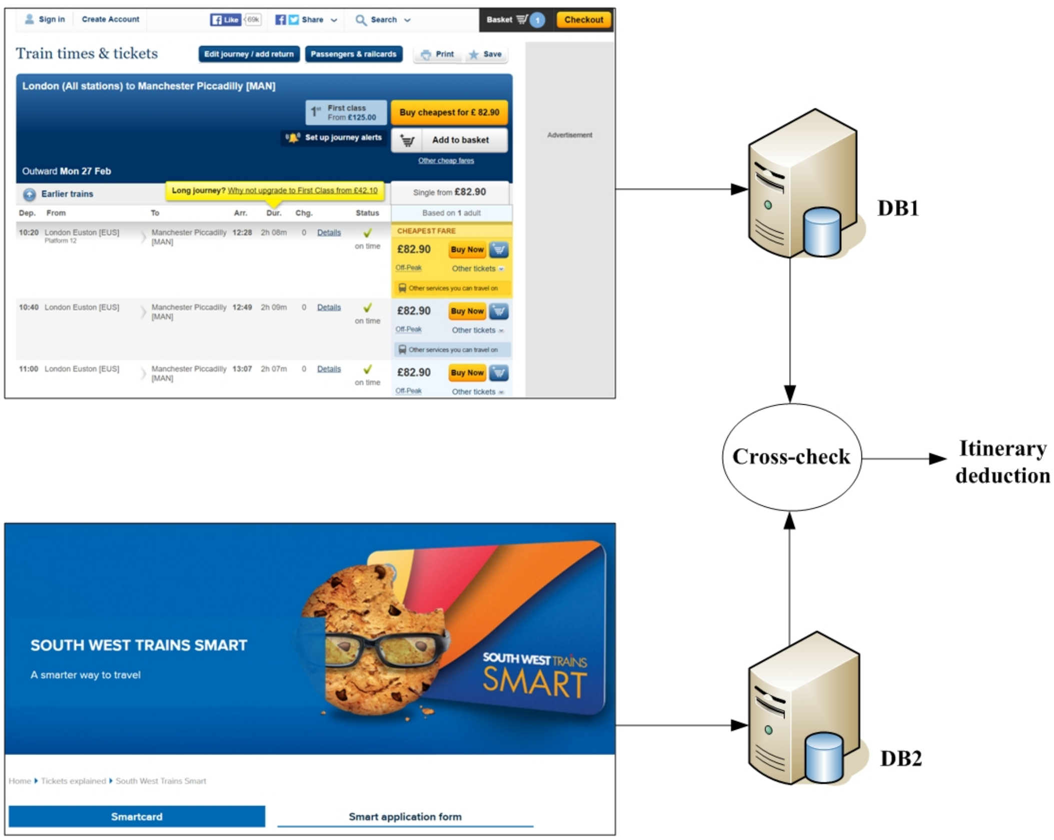


Figure 5.21 An outlook for future passenger-oriented rail research

The central idea of Figure 5.21 can be briefly explained as follows. Firstly, two databases (denoted by DB1 and DB2 in the figure) need to be set up: one (i.e. DB1) is used to store information about the click events corresponding to pre-planned/recommended itineraries; and the other (i.e. DB2) is used to store information about transaction events corresponding to fare payments at railway stations. Then, a sufficiently large sample needs to be extracted from each of the two databases to make comparisons and analyses according to some predefined filtration rules to deduct itinerary-specific passenger flows.

To implement the above 'blueprint' for more detailed and passenger-perspective studies, two potential obstacles should firstly be overcome. The first potential obstacle lies in technical feasibility: to accurately identify and record each effective click would be a challenging task, considering the huge daily traffic of NRE[20]. The other obstacle lies mainly in the coordination of different rail operators: since the Smartcard scheme is still at an early stage, only part of the train operating companies in Britain have participated in this scheme and a particular smart card is largely restricted to operator-specific routes and stations, which is far from able to cover those long-distance and transfer-involved routes at the time of writing this thesis.

In a word, the development of software solutions and that of hardware solutions are interdependent: the potential of software solutions can be fully realised only if the relevant hardware technology could 'catch up', and vice versa.

5.8 Conclusions

This chapter has been mainly focused on the description of two novel route-level measures developed to quantify the effect of modified pre-trip information. Generally speaking, the introduction of the two route-level measures and the corresponding analytical methods can serve the following three purposes: 1) enables empirical analyses of those identified critical routes (presented in Chapter 3) using detailed data about historical train movements; 2) provides a way to evaluate the effectiveness of the proposed algorithmic approach (presented in Chapter 4) in coping with those identified critical routes; and 3) provides a reference point for more detailed microscopic analyses of those transfer-involved rail routes.

More specifically, RPM (Route-oriented Performance Measure) is developed to evaluate the overall performance of a given transfer-involved rail route in terms of punctuality and reliability during a given observation period. The RPM-based analytical method is mainly

[20] http://www.nationalrail.co.uk/46383.aspx

built on an assumption of an 'average' passenger and an assumption of an absolute standard/threshold for determining whether an 'average' passenger has been delayed, which can be viewed as an extension of PPM (Public Performance Measure), the industry standard adopted in British railways. Based on a detailed explanation of the underlying sample data and the specific technicalities used, RPM-based analyses of nine identified critical routes in Britain's passenger rail system have been conducted. The obtained evaluation results reveal that the modified pre-trip information generated from the algorithmic approach proposed in Chapter 4 could, clearly, bring improvements in punctuality and reliability to these identified critical routes. The potential limitations of RPM and the algorithmic approach proposed in Chapter 4 have also been recognised by an in-depth investigation into the sample data about several representative routes: like PPM, RPM tends to be unable to reflect the whole picture of the underlying sample data, indicating the necessity of auxiliary performance indicators such as CaSL (Cancellation and Significant Lateness); the modified pre-trip information generated from the algorithmic approach proposed in Chapter 4 mainly covers those medium-sized delays but has little effect on those significant delays.

As an exploration of a more comprehensive measurement incorporating more realistic scenarios in route and itinerary choice, another route-level measure is developed called RUM (Route-oriented Utility Measure). RUM deserts the assumptions of an 'average' passenger and an absolute standard for distinguishing between lateness and punctuality. Instead, it takes into account the heterogeneity among rail passengers and measures the change the modified pre-trip information could have brought in the overall utility of the relevant passengers choosing a particular route (during a given observation period). In order to implement an RUM-based analysis, several 'bold' assumptions need to be introduced such as an assumption of the existence of 'latest-tolerable arrival times' (LATs) and an assumption of the existence of 'unwary' passengers. Moreover, the principle of indifference (POI) is implicitly included in the RUM-based analytical model due to a lack of detailed data about passenger flows along these transfer-involved rail routes. Based on a detailed explanation of the underlying sample data and the specific technicalities used, RUM-based analyses of nine identified critical routes in Britain's passenger rail system have been conducted. The possible reasons for several 'abnormal' results have been analysed by a close examination of the underlying sample data and the mechanism of the proposed analytical models. Generally, these evaluation results have enriched our understanding of these identified critical routes. Although the RUM-based analytical method is largely experimental, it could easily be extended to conduct more realistic microscopic analyses of those transfer-involved rail routes, as long as detailed data about passenger flows and passenger activities become available.

Chapter 6
Conclusions

6.1 Brief summary

Passenger rail transport is one of the major alternatives to car transport in many European countries such as Britain. However, the national railway network in Britain is becoming more and more crowded and prone to small delays and major disruptions, due to an ever-increasing demand for passenger rail transport over the last two decades. One of the negative effects of a delay-prone railway network is on those transfer-involved rail journeys, due to increased exposure to missed transfers. Conventional solutions to a delay-prone railway network are either costly and time-consuming (e.g. an extensive upgrading of rail infrastructure), or unable to allow for the diverse realistic scenarios in passenger rail transport (e.g. timetable design at the tactical planning phase). Observing that advanced passenger information systems (e.g. passenger information websites/mobile apps, departure boards within stations, etc) have been playing an increasingly important role in passengers' experience of rail services in the developed world, this thesis tries to develop an information-based solution to the problem of delay and disruption management to deal with those blind spots over which existing solutions have little control.

Of particular interest to this thesis are those transfer-involved rail routes, which have received relatively less attention from rail operators compared with direct routes, not only in terms of timetable design but also in terms of pre-trip passenger information. In order to formulate the problem of pre-trip information about those transfer-involved routes, three novel concepts – critical transfers, critical itineraries, and critical routes – are introduced. Roughly speaking, a critical itinerary is composed of critical transfers, each of which is delay-sensitive and is associated with high consequence if missed. And if the recommended itinerary list (by a journey planning system) is full of critical itineraries, the corresponding route would be problematic in terms of punctuality and reliability and is called a critical route.

An efficient screening algorithm, named Critical Routes Finder (CRF), is developed and implemented to check whether there exist critical routes within a given railway system and to find out, if existent, which of those transfer-involved routes are critical. The screening algorithm is then applied to analyse the current National Rail timetable (valid between 11 December 2016 and 20 May 2017) adopted by British railways to identify those critical routes within Britain's passenger rail system. The performance of the screening algorithm is promising in terms of computational efficiency. The screening results show that more attention should be paid to such transfer-involved routes as London Kings Cross – York – Scarborough, Bournemouth – Southampton Central – Brighton, etc to improve the pre-trip information about these routes.

In order to find, from within the domain of information technology itself, a solution to the problem of pre-trip information about those identified critical routes, a brief review of the relevant prototypes in the literature and the relevant applications in the real world has been conducted: it is recognised that the existing information-related approaches have not truly touched upon the problem of critical routes, either in theory or in practice. But these existing approaches can be utilised as building blocks to develop a solution to the problem of critical routes.

Inspired by some existing travel information technologies, a historical-data-based approach is developed, containing a series of easy-to-implement algorithms. The design philosophy behind the algorithmic approach proposed is a 'local treatment' of those identified critical routes (rather than a 'holistic treatment' of all possible routes within a railway network), which differs from the various existing approaches. This different treatment could significantly reduce computational complexity and meanwhile avoids disturbing information about those non-critical routes.

Three interrelated algorithms are proposed and detailed, which are named IPS, PBPM, and PBPM+, respectively. IPS (Itinerary-oriented Performance Statistics) has been inspired by those individual-leg-oriented performance statistics accessible from some existing travel information websites. Roughly speaking, IPS can be viewed as an augmented version of those individual-leg-oriented performance statistics: it is designed to compute and present performance statistics that are oriented toward a whole journey (itinerary) rather than toward individual service legs, which would be able to set the information consumers (passengers) free from reprocessing the fragmented information (about individual legs) by themselves.

Despite their advantage over individual-leg-oriented performance statistics, itinerary-oriented performance statistics may still make information consumers (passengers) feel like they are gambling and hence cause inconvenience/uneasiness to them. Based on such a consideration, PBPM (Performance-Based Pre-Modification of advertised arrival times) is developed. PBPM has been inspired by the relevant technologies in real-time delay information: it abandons the output of performance statistics; instead, it consumes performance statistics as intermediate results to compute the final results – pre-modified (advertised) arrival times – well before the time of travel. Roughly speaking, a pre-modified (advertised) arrival time of a given critical itinerary reflects the 'average lateness' of this itinerary over the last several weeks, incorporating both the risk of missed transfers (reliability) and the average delay at the destination station (punctuality).

Although the final results of PBPM can be readily delivered to end users (passengers) for enhanced pre-trip information, these results (i.e. pre-modified arrival and journey times) are still largely descriptive: for those passengers having a relatively tight schedule, they would still have no alternative choices when the available options (i.e. recommended itineraries) are found to be undesirable. Based on such a consideration, PBPM+ is developed, the purpose of which is to further extend the functionality of PBPM to generate additional prescriptive information about alternative itineraries when necessary. Roughly speaking, PBPM+ incorporates the results obtain from PBPM into existing journey planning algorithms to influence journey planning results. More specifically, this can be achieved by modifying the relevant parameters of a journey planning algorithm and adding to the algorithm additional post-processing procedures.

In order to evaluate the effectiveness of the information-based solution to the problem of critical routes, two novel route-level measures are developed and detailed. Generally speaking, the introduction of the two route-level measures and the corresponding analytical methods can serve the following three purposes: 1) enables empirical analyses of those identified critical routes presented in Chapter 3 using detailed data about historical train movements; 2) provides a way to evaluate the effectiveness of the proposed algorithmic approach presented in Chapter 4 in coping with those identified critical routes; and 3) provides a reference point for more detailed microscopic analyses of those transfer-involved rail routes.

More specifically, RPM (Route-oriented Performance Measure) is developed to evaluate the overall performance of a given transfer-involved rail route in terms of punctuality and reliability during a given observation period. The RPM-based analytical method is mainly built on an assumption of an 'average' passenger and an assumption of an absolute standard/threshold for determining whether an 'average' passenger has been delayed, which can be viewed as an extension of PPM (Public Performance Measure), the industry standard adopted in British railways. Based on a detailed explanation of the underlying sample data and the specific technicalities used, RPM-based analyses of nine identified critical routes in Britain's passenger rail system have been conducted. The obtained evaluation results reveal that the modified pre-trip information generated from the approach proposed in Chapter 4 could clearly bring improvements in punctuality and reliability to these identified critical routes. The potential limitations of RPM and the algorithmic approach proposed have also been recognised by an in-depth investigation into the sample data about several representative routes: like PPM, RPM tends to be unable to reflect the whole picture of the underlying sample data, indicating the necessity of auxiliary performance indicators such as CaSL (Cancellation and Significant Lateness); and the modified pre-trip information generated from the approach proposed in Chapter 4 mainly covers those medium-sized delays but has little effect on those significant delays.

As an exploration of a more comprehensive measurement incorporating more realistic scenarios in route and itinerary choice, another route-level measure is developed, named RUM (Route-oriented Utility Measure). RUM abandons the assumptions of an 'average' passenger and an absolute standard for determining whether an 'average' passenger has been delayed. Instead, it takes into account the heterogeneity among rail passengers and measures the change the modified pre-trip information could have brought in the overall utility of the relevant passengers choosing a particular route (during a given observation period). In order to implement an RUM-based analysis, several 'bold' assumptions need to be introduced such as an assumption of the existence of 'latest-tolerable arrival times' (LATs) and an assumption of the existence of 'unwary' passengers. Moreover, the principle of indifference (POI) is implicitly included in the RUM-based analytical model due to a lack of detailed data about passenger flows on these transfer-involved rail routes. Based on a detailed explanation of the underlying sample data and the specific technicalities used, RUM-based analyses of nine identified critical routes in Britain's passenger rail system have been conducted. The possible reasons for several 'abnormal' results have been analysed by a close examination of the underlying sample data and the mechanism of the proposed analytical models. Generally, these evaluation results have enriched our understanding of these identified critical routes. Although the RUM-based analytical method is largely experimental, it could easily be extended to conduct more realistic microscopic analyses of those transfer-involved rail routes, as long as detailed data about passenger flows and passenger activities become available.

6.2 Main findings

By reviewing the existing theories and applications in railway planning and passenger information in Chapters 2 and 3, it is recognised that the pre-trip information about those transfer-involved rail routes may be a potential problem: due to the inherent defects in railway timetabling and journey planning technologies, the quality of the pre-trip information about those transfer-involved rail routes cannot always be guaranteed.

In Chapter 3, an in-depth analysis is conducted of a quite large sample of train movements data. It is found that train delays in British railways can be better modelled by those compound distributions (than those 'pure' distributions), among which q-exponential models tend to be the most promising candidate in terms of the overall goodness of fit. Moreover, by comparing the best-fit q-exponential model of the latest train delay data with that of the 2005/06 data, a noticeable increase in small-sized delays (from one to eight minutes) has been identified in British railways over the past decade.

From the detailed descriptions and explanations of the proposed algorithms and analytical methods in Chapters 4 and 5, it can be seen that open data available from Britain's rail industry contain a lot of details about daily train movements, which can be exploited to conduct some microscopic analyses of those transfer-involved rail routes.

The empirical results presented in Chapter 5 reveal that the algorithmic approach of using historical train movements data to pre-modify recommended itineraries can largely resolve the problem of the pre-trip information about those transfer-involved rail routes, although its effectiveness cannot be guaranteed in all cases. Specifically, the proposed algorithmic approach can bring an average gain of 5.0% ~ 11.3% in terms of a train-oriented performance measure (named RPM) to the nine studied critical routes in British railways, and can bring an average gain of 2.8% in terms of a passenger-oriented utility measure (named RUM) to six of the nine studied critical routes. Three of the nine studied routes cannot gain in RUM, although they can gain in RPM.

6.3 Methodological contributions

A relatively comprehensive survey is conducted of the state-of-the-art theories and technologies of several different disciplines (c.f. Chapters 2 – 4), potentially facilitating the interested researchers to make more contributions to the solution of the relevant issues.

A screening algorithm (i.e. CRF in Chapter 3) is developed to efficiently locate those weak points (i.e. critical routes) within a national-level railway system, which provides an additional tool/option for timetable design and analysis.

A set of three interrelated information enhancing algorithms (i.e. IPS, PBPM, and PBPM+) are developed to cope with those weak points (i.e. critical routes) within a national-level railway system, among which the central idea of PBPM – floating (variable/adjustable) allowances – provides a potentially useful (additional) tool for delay management in railway timetabling and operations.

A route-level performance measure (i.e. RPM) is developed by augmenting the current industry standard (i.e. PPM), which can be utilised to conduct route-level evaluations and comparisons for those transfer-involved rail routes (in terms of punctuality and reliability). Apart from the ability to quantify the effect of a specific information enhancing strategy in the context of this thesis, RPM may also be readily employed to evaluate and compare the performances of those direct rail routes (lines).

A route-level utility measure (i.e. RUM) is developed that takes into account passenger delays. Similar to RPM, RUM's applicability is not limited to those quantitative analyses in the specific context of this thesis: it has broader applicability in a variety of potential applications such as employing RUM as an additional optimisation criterion in the timetabling phase to reflect passenger interests, using RUM to check whether the allocation of allowances (time supplements) is effective enough, and integrating RUM into the historical-data-based algorithmic approach proposed in this thesis to augment the original version (c.f. Section 5.6).

Figure 6.1 below provides a graphical description of the potential contributions of this thesis to railway timetabling and operations: the relevant algorithms and analytical methods described in this thesis can not only be integrated into one framework to improve the pre-trip information about those transfer-involved rail routes, but also be applied separately to different processes in railway timetabling and operations to achieve different goals.

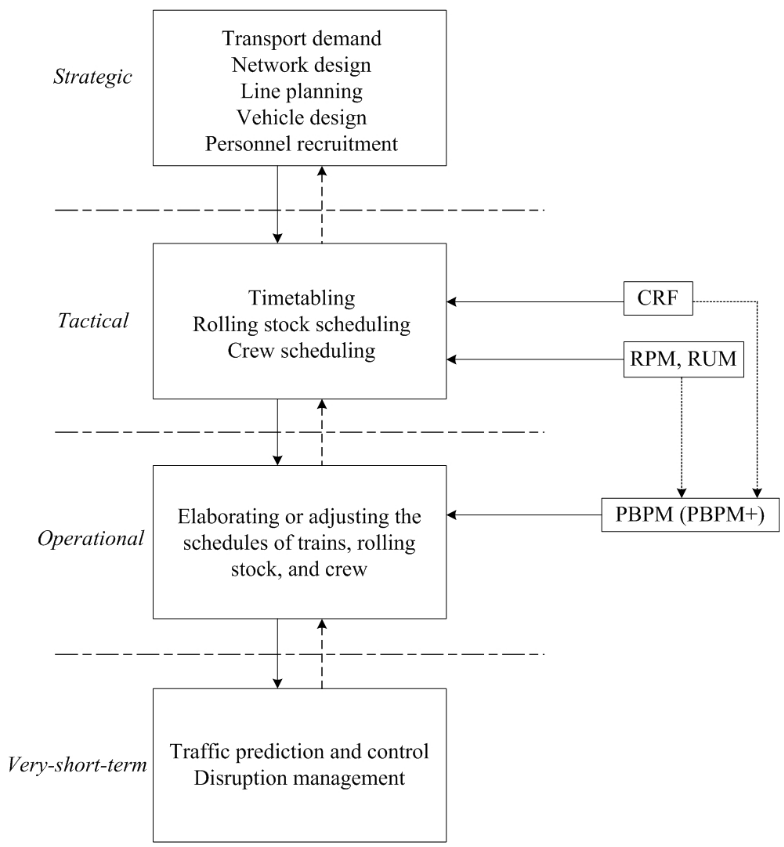


Figure 6.1 An illustration of the potential contributions of this thesis to railway timetabling and operations

6.4 Contributions to knowledge

The explanations, illustrations, and abstractions of critical routes, mainly embodied by Sections 3.4 and 3.5, could be viewed as an original contribution to knowledge – an unresolved problem existing in current pre-trip timetable information systems has been revealed.

The statistical results and stochastic models obtained based on real-world train operation records, embodied by Sections 3.7 and 3.8, could help better understand and update the knowledge of the macro-level delay distributions within Britain's passenger rail system.

The categorisation of the various existing systems, prototypes, and algorithmic ideas, embodied by Section 4.2, could also be regarded as an original contribution to knowledge, for there exists no such categorisation in the large body of relevant literature.

Those route-level analyses and assessments conducted based on real-world train operation records, embodied by Section 5.3 and Section 5.5, would help any interested reader learn, on a variety of dimensions, about those critical routes within Britain's passenger rail system.

6.5 Limitations and future research

Several identified (potential) limitations of the proposed algorithmic solutions and analytical methods have been analysed/explained in Section 4.5 and Section 5.7 of this thesis. In short, the relatively large granularity (i.e. precision tolerance) of the available train movements data and the lack of detailed data about passenger counts and passenger flows may to some degree restrict the precision and deepness of the relevant evaluations and analyses.

Based on the identified limitations and imperfections in this thesis, four conceivable directions for further research are recommended below.

Firstly, further improve the information enhancing algorithmic approach proposed in this thesis. From Chapter 5, it can be seen that although the proposed (historical-data-based) algorithmic approach can largely improve the pre-trip information about those critical routes, it is not perfect: empirical results indicate that there exist some (if not many) exceptions. A typical exception in the analyses presented in Chapter 5 is the route Ebbw Vale Town – Cardiff Central – Birmingham New Street (i.e. ECB): its arrival delays seem to be largely unpredictable by straightforward statistics (i.e. the proposed algorithmic approach) and hence effective solutions to these exceptions may be needed. Machine learning is a conceivable path towards dealing with these ‘exceptions’, but empirical evidence is needed to prove or disprove its effectiveness.

Secondly, devise more realistic quality measure(s): the results presented in Chapter 5 have partly shown the potential limitations of the currently adopted industry standard (i.e. PPM). Generally speaking, the current standard is largely train-oriented (rather than passenger-oriented) and does not take into account a number of realistic factors (e.g. passenger flows, the heterogeneity in perceptions of delays, etc). In the future, more realistic measure(s) can be introduced as long as the relevant data become available.

Thirdly, use big data to gain more knowledge about the mechanism of train delays in British railways. In Section 3.8 of this thesis, statistical analyses of the train delays in British railways have been conducted using a relatively large sample (about 1.4 million records) of historical train movements data. However, the obtained results are largely synthetic/aggregate, from which only general conclusions can be drawn. Several important questions remain unanswered such as what the underlying mechanism is of those extremely large delays and how train delays in British railways are distributed on the dimensions of time and space. The reason why these questions have not been touched upon is mainly due to the fact that the adopted sample is still a small and potentially biased sample (corresponding only to 14 stations for 12 months) and its representativeness remains dubious. In the future, these unanswered questions may be able to be confidently answered, once a truly large and representative sample becomes available.

Fourthly, stated preference (SP) studies (c.f. Kroes and Sheldon, 1988) can be conducted in the future to monetise the relevant utility indices. Although this thesis does not involve monetised utilities and SP methods themselves have been questioned in the literature (e.g.

Diamond and Hausman, 1994), monetised utilities are still useful tools for strategic railway planning in the foreseeable future.

Bibliography

Adler, M.D. (2014) Extended Preferences and Interpersonal Comparisons: A New Account. *Economics and Philosophy*, 30, 123-162.

Adler, M.D. and Fleurbaey, M. (2016) *The Oxford Handbook of Well-Being and Public Policy*. Oxford University Press.

Allulli, L., Italiano, G.F., and Santaroni, F. (2014) Exploiting GPS Data in Public Transport Journey Planners. In *Proceedings of the 13th International Symposium on Experimental Algorithms (SEA'14)*, Lecture Notes in Computer Science, 8504, 295-306.

Andersson, E.V. (2014) *Assessment of Robustness in Railway Traffic Timetables*. Licentiate Thesis, Linköping University.

Andersson, E.V., Peterson, A. and Krasemann, J.T. (2013) Quantifying railway timetable robustness in critical points. *Journal of Rail Transport Planning & Management*, 3(3), 95-110.

Armstrong, J. and Preston, J. (2017) Capacity utilisation and performance at railway stations. *Journal of Rail Transport Planning & Management*, 7(3), 187-205.

ATOC (Association of Train Operating Companies) (2013) *Passenger Demand Forecasting Handbook*. London: Association of Train Operating Companies.

ATOC (2016) National Rail Data Portal. https://datafeeds.nationalrail.co.uk/ [Accessed 20 November 2016]

Bailey, C. (1995) *European Railway Signalling*. A & C Black, London.

Bast, H., Carlsson, E., Eigenwillig, A., Geisberger, R., Harrelson, C., Raychev, V. and Viger, F. (2010) Fast routing in very large public transportation networks using transfer patterns. In *Proceedings of the 18th Annual European Symposium on Algorithms (ESA'10)*, *Lecture Notes in Computer Science*, 6346, 290-301.

Bast, H., Delling, D., Goldberg, A.V., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D. and Werneck, R.F. (2015) *Route Planning in Transportation Networks*. Technical Report, Microsoft Research, an updated version of MSR-TR-2014-4.

Bast, H. and Storandt, S. (2014) Frequency-based search for public transit. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '14)*, 13-22.

Bates, J., Polak, J., Jones, P. and Cook, A. (2001) The valuation of reliability for personal travel. *Transportation Research Part E*, 37(2), 191-229.

Ben-Elia, E. and Avineri, E. (2015) Response to Travel Information: A Behavioural Review. *Transport Reviews*, 35(3), 352-377.

Ben-Elia, E., Di Pace, R., Bifulco, G.N. and Shiftan, Y. (2013) The impact of travel information's accuracy on route-choice. *Transportation Research Part C*, 26, 146-159.

Berger, A., Delling, D., Gebhardt, A. and Müller-Hannemann, M. (2009) Accelerating TimeDependent Multi-Criteria Timetable Information is Harder Than Expected. In *Proceedings of the 9th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'09)*, OpenAccess Series in Informatics (OASIcs): 1-21.

Bergström, A. and Krüger, N.A. (2013) *Modeling Passenger Train Delay Distributions – Evidence and Implications*. CTS Working Paper 2013:3, Centre for Transport Studies, Sweden.

Böhmová, K., Mihalák, M., Neubert, P., Pröger, T. and Widmayer, P. (2015) Robust Routing in Urban Public Transportation: Evaluating Strategies that Learn From the Past. In *Proceedings of the 15th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'15)*, OpenAccess Series in Informatics (OASIcs): 68-81.

Böhmová, K., Mihalák, M., Pröger, T., Šrámek, R. and Widmayer, P. (2013) Robust routing in urban public transportation: How to find reliable journeys based on past observations. In *Proceedings of the 13th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'13)*, OpenAccess Series in Informatics (OASIcs): 27-41.

Bradley, R. (2014) *Decision Theory: A Formal Philosophical Introduction.* Available from: http://personal.lse.ac.uk/bradleyr/pdf/Handbook%20%20Decision%20theory%20(revised).pdf [Accessed 30 November 2016]

Briggs, K. and Beck, C. (2007) Modelling train delays with q-exponential functions. *Physica A*, 378, 498-504.

Briggs, R. (2017) "Normative Theories of Rational Choice: Expected Utility", *The Stanford Encyclopedia of Philosophy* (Spring 2017 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/spr2017/entries/rationality-normative-utility/>.

Brodal, G. and Jacob, R. (2004) Time-dependent networks as models to achieve fast exact time-table queries. In *Proceedings of the 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'03)*, *Electronic Notes in Theoretical Computer Science*, 92, 3-15.

Caimi, G., Kroon, L. and Liebchen, C. (2017) Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, 6(4), 285-312.

Carrasco, N. (2012) Quantifying reliability of transit service in Zurich, Switzerland. *Transportation Research Record*, 2274, 114-125.

Chorus, C.G., Arentze, T.A. and Timmermans, H.J.P. (2007) Information impact on quality of multimodal travel choices: conceptualizations and empirical analyses. *Transportation*, 34, 625-645.

Cionini, A., D'Angelo, G., D'Emidio, M., Frigioni, D., Giannakopoulou, K., Paraskevopoulos, A. and Zaroliagis, C. (2014) Engineering Graph-Based Models for Dynamic Timetable Information Systems. In *Proceedings of the 14th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'14)*, OpenAccess Series in Informatics (OASIcs): 46-61.

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2009) *Introduction to Algorithms*, 3rd ed. Cambridge: The MIT Press.

DAB (Delay Attribution Board) (2016) *Delay Attribution Guide - September 2016*. Available from: http://www.delayattributionboard.co.uk/documents/dag_pdac/Current%20Delay%20Attribution%20Guide.pdf [Accessed 20 November 2016]

D'Ariano, A., Corman, F. and Pacciarelli, D. (2014) Rescheduling, in: I.A. Hansen and J. Pachl (Eds) *Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation.*, pp. 253-273.

Dean, B.C. (1999) *Continuous-time dynamic shortest path algorithms*. Master Thesis, Massachusetts Institute of Technology.

Delling, D., Katz, B. and Pajor, T. (2012) Parallel computation of best connections in public transportation networks. *ACM Journal of Experimental Algorithmics*, 17(4), 1-26.

Delling, D., Pajor, T., and Wagner, D. (2009) Engineering time-expanded graphs for faster timetable information. In *Robust and Online Large-Scale Optimization*, *Lecture Notes in Computer Science*, 5868, 182-206.

Delling, D., Italiano, G.F., Pajor, T. and Santaroni, F. (2014a) Better Transit Routing By Exploiting Vehicle GPS Data. *In Proceedings of the 7th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. ACM Press, November 2014.

Delling, D., Pajor, T. and Werneck, R.F. (2014b) Round-based public transit routing. *Transportation Science*, 49(3), 591-604.

Delling, D. and Wagner, D. (2009) Time-dependent route planning. In *Robust and Online Large-Scale Optimization*, *Lecture Notes in Computer Science*, 5868, 207-230.

Demeyer, S. (2013) *Innovative Algorithms for the Planning and Routing of Multimodal Transportation*. PhD Thesis, Ghent University.

DfT (Department for Transport) (2013) *National Rail Travel Survey – Overview report*. Available from: https://www.gov.uk/government/statistics/national-rail-travel-survey-overview-report [Accessed 20 November 2016]

DfT (2016a) *National Travel Survey: 2015*. Available from: https://www.gov.uk/government/statistics/national-travel-survey-2015 [Accessed 20 November 2016]

DfT (2016b) *Rail factsheets: 2016*. Available from: https://www.gov.uk/government/statistics/rail-factsheets-2016 [Accessed 03 December 2016]

DfT(2017) *Statistical data set: Rail usage, infrastructure and performance (RAI01).* Available from: https://www.gov.uk/government/statistical-data-sets/rai01-length-of-route-distance-travelled-age-of-stock [Accessed 02 March 2017]

Diamond, P.A. and Hausman, J.A. (1994) Contingent Valuation: Is Some Number Better than No Number? *Journal of Economic Perspectives*, 8(4), 45-64.

Dibbelt, J., Pajor, T., Strasser, B. and Wagner, D. (2013) Intriguingly Simple and Fast Transit Routing. In *Proceedings of the 12th International Symposium on Experimental Algorithms (SEA'13)*, *Lecture Notes in Computer Science*, 7933, 43-54.

Dibbelt, J., Strasser, B. and Wagner, D. (2014) Delay-Robust Journeys in Timetable Networks with Minimum Expected Arrival Time. In *Proceedings of the 14th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'14)*, OpenAccess Series in Informatics (OASIcs): 1-14.

Diestel, R. (2010) *Graph Theory*, 4th ed. Springer-Verlag, Heidelberg.

Disser, Y., Müller-Hannemann, M. and Schnee, M. (2008) Multi-criteria shortest paths in time-dependent train networks. In *Proceedings of the 7th Workshop on Experimental Algorithms (WEA'08)*, *Lecture Notes in Computer Science*, 5038, 347-361.

de Elia, R. and Laprise, R. (2005) Diversity in Interpretations of Probability: Implications for Weather Forecasting. *Monthly Weather Review*, 133(5), 1129-1143.

Farag, S. and Lyons, G. (2008) What affects use of pretrip public transport information? Empirical results of a qualitative study. *Transportation Research Record*, 2069, 85-92.

Farag, S. and Lyons, G. (2012) To use or not to use? An empirical study of pre-trip public transport information for business and leisure trips and comparison with car travel. *Transport Policy*, 20, 82-92.

Fujiyama, T. and Tyler, N. (2010) Predicting the walking speed of pedestrians on stairs. *Transportation Planning and Technology*, 33(2), 177-202.

Goerigk, M., Heße, S., Müller-Hannemann, M. and Schmidt, M. (2013) Recoverable robust timetable information. In *Proceedings of the 13th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'13)*, OpenAccess Series in Informatics (OASIcs): 1-14.

Goerigk, M., Knoth, M., Müller-Hannemann, M., Schmidt, M. and Schöbel, A. (2014) The price of strict and light robustness in timetable information. *Transportation Science*, 48(2), 225-242.

Gordon, J.B., Koutsopoulos, H.N., Wilson, N.H.M. and Attanucci, J.P. (2014) Automated Inference of Linked Transit Journeys in London Using Fare-Transaction and Vehicle Location Data. *Transportation Research Record*, 2343, 17-24.

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

Goverde, R.M.P. (2014) Timetable Stability Analysis, in: I.A. Hansen and J. Pachl (Eds) *Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation.*, pp. 133-153.

Grinstead, C.M. and Snell, J.L. (1997) *Introduction to Probability*, 2nd ed. American Mathematical Society.

Guo, Y. and Preston, J. (2016) Improving pre-trip information in Britain's passenger rail system. Paper presented at 48th Annual UTSG Conference, Bristol, 6-8 January.

Guo, Z. (2008) *Transfers and Path Choice in Public Transport Systems*. PhD Thesis, Massachusetts Institute of Technology.

Hájek, A. (2012) "Interpretations of Probability", *The Stanford Encyclopedia of Philosophy* (Winter 2012 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/win2012/entries/probability-interpret/>.

Hansen, P. (1979) Bricriteria path problems. In *Multiple Criteria Decision Making – Theory and Application* –, 109-127.

Hansson, S.O. (1994) *Decision Theory: A Brief Introduction*. Available from: http://people.kth.se/~soh/decisiontheory.pdf [Accessed 30 November 2016]

Harris, N.G. (1992) Punctuality and Performance, in: N.G. Harris and E.W. Godward (Eds) *Planning Passenger Railways.*, pp. 130-142.

Harris, N.G., Mjøsund, C.S. and Haugland, H. (2013) Improving railway performance in Norway. *Journal of Rail Transport Planning & Management*, 3, 172-180.

Harsanyi, J.C. (1997) Utilities, preferences, and substantive goods. *Social Choice and Welfare*, 14, 129-145.

Huisman, D., Kroon, L.G., Lentink, R.M. and Vromans, M.J.C.M. (2005) Operations Research in passenger railway transportation. *Statistica Neerlandica*, 59(4), 467-497.

van der Hurk, E. (2015) *Passengers, Information, and Disruptions*. PhD Thesis: Erasmus University Rotterdam.

Kahneman, D. (2012) *Thinking, fast and slow.* New York, NY: Farrar, Straus and Giroux.

Kamm, F. (1998) Précis of Morality, Mortality, Vol. 1: Death and Whom to Save from It. *Philosophy and Phenomenological Research*, 58(4), 939-945.

Keyhani, M.H. (2017) *Computing Highly Reliable Train Journeys*. PhD Thesis: Technischen Universität Darmstadt.

Kroes, E.P. and Sheldon, R.J. (1988) Stated Preference Methods: An Introduction. *Journal of Transport Economics and Policy*, 22(1), 11-25.

Kroon, L., Huisman, D. and Maróti, G. (2014) Optimisation Models for Railway Timetabling, in: I.A. Hansen and J. Pachl (Eds) *Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation.*, pp. 155-173.

Kurosaki, F. (2008) *An Analysis of Vertical Separation of Railways*. PhD Thesis: The University of Leeds.

Landex, A. (2008) *Methods to estimate railway capacity and passenger delays*. PhD Thesis: Technical University of Denmark.

Lyons, G. (2015) Transport's digital age transition. *The Journal of Transport and Land Use*, 8(2), 1-19.

Lyons, G., Avineri, E. and Farag, S. (2008) Assessing the demand for travel information: Do we really want to know? In: *European Transport Conference*, Leeuwenhorst Conference Centre, The Netherlands, 6-8 October.

Mahmassani, H.S. and Chang, G.L. (1986) Experiments with Departure Time Choice Dynamics of Urban Commuters. *Transportation Research Part B*, 20B, 297-320.

Martin, U. (2014) Performance Evaluation, in: I.A. Hansen and J. Pachl (Eds) *Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation.*, pp. 275-293.

Meeker, M. (2016) *Internet trends 2016- Code Conference*. KPCB's annual Internet Trends Report. http://www.kpcb.com/internet-trends [Accessed 20 November 2016]

Mellouli, T. and Suhl, L. (2006) Passenger online routing in dynamic networks. In D. C. Mattfeld and L. Suhl (Eds) *Informationsprobleme in Transport und Verkehr*, 4, 17-30.

Mizutani, F., Smith, A., Nash, C. and Uranishi, S. (2014) *Comparing the Costs of Vertical Separation, Integration, and Intermediate Organisational Structures in European and East Asian Railways*. Discussion Paper Series 2014-37: Graduate School of Business Administration, Kobe University.

Mood, A.M. (1974) *Introduction to the Theory of Statistics, 3rd ed*. McGraw Hill.

Müller-Hannemann, M. and Schnee, M. (2007) Finding all attractive train connections by multi-criteria Pareto search. In *Algorithmic Methods for Railway Optimization*, *Lecture Notes in Computer Science*, 4359, 246-263.

Müller-Hannemann, M. and Schnee, M. (2009) Efficient timetable information in the presence of delays. In *Robust and Online Large-Scale Optimization*, Lecture Notes in Computer Science, 5868, 249-272.

Müller-Hannemann, M. and Weihe, K. (2001) Pareto shortest paths is often feasible in practice. In *Proceedings of the 5th International Workshop on Algorithm Engineering (WAE'01)*, *Lecture Notes in Computer Science*, 2141, 185-197.

Nachtigall, K. (1995) Time depending shortest-path problems with applications to railway networks. *European Journal of Operational Research*, 83(1), 154-166.

Network Rail (2016a) *Timetabling explained.* Available from: http://www.networkrail.co.uk/timetables-and-travel/timetabling-explained/ [Accessed 20 November 2016]

Network Rail (2016b) *Electronic national rail timetable*. Available from: https://www.networkrail.co.uk/running-the-railway/timetabling/electronic-national-rail-timetable-enrt/ [Accessed 20 November 2016]

Network Rail (2017) *Public Performance Measure*. Available from: https://www.networkrail.co.uk/who-we-are/how-we-work/performance/public-performance-measure/ [Accessed 15 January 2017]

Ng, YK. (1997) A Case for Happiness, Cardinalism, and Interpersonal Comparability. *The Economic Journal*, 107(445), 1848-1858.

Noland, R.B. and Polak, J.W. (2002) Travel time variability: A review of theoretical and empirical issues. *Transport Reviews*, 22, 39-54.

Olsson, N.O.E. and Haugland, H. (2004) Influencing factors on train punctuality – results from some Norwegian studies. *Transport Policy*, 11, 387-397.

ONS (2016) *Internet Access – Households and Individuals: 2016*. Statistical Bulletin. Office for National Statistics. https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/bulletins/internetaccesshouseholdsandindividuals/2016 [Accessed 20 November 2016]

ORR (Office of Rail and Road) (2016) *Passenger Rail Usage 2015-16 Q4 Statistical Release*. Available from: http://orr.gov.uk/statistics/published-stats/statistical-releases [Accessed 25 June 2016]

Orda, A. and Rom, R. (1990) Shortest-path and minimum delay algorithms in networks with timedependent edge-length. *Journal of the ACM*, 37(3), 607-625.

Orda, A. and Rom, R. (1991) Minimum weight paths in time-dependent networks. *Networks*, 21, 295-319.

ORDW (Open Rail Data wiki) (2016a) *List of Train Describers*. Available from: http://nrodwiki.rockshore.net/index.php/List_of_Train_Describers, [Accessed 20 Nov 2016]

ORDW (2016b) *Train Movements*. Available from: http://nrodwiki.rockshore.net/index.php/Train_Movements, [Accessed 20 Nov 2016]

Pachl, J. (2014) Timetable Design Principles, in: I.A. Hansen and J. Pachl (Eds) *Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation.*, pp. 13-46.

Papoulis, A. (1991) *Probability, Random Variables, and Stochastic Processes, 3rd ed*. McGraw Hill.

Parbo, J., Nielsen, O.A. and Prato, C.G. (2016) Passenger Perspectives in Railway Timetabling: A Literature Review. *Transport Reviews*, 36(4), 500-526.

Parmigiani, G. and Inoue, L. (2009) *Decision Theory: Principles and Approaches*. Wiley.

Pelletier, M.P., Trépanier, M. and Morency, C. (2011) Smart card data use in public transit: A literature review. *Transportation Research Part C*, 19, 557-568.

Porter, A.L. and Rafols, I. (2009) Is science becoming more interdisciplinary? Measuring and mapping six research fields over time. *Scientometrics* , 81(3), 719-745.

Preston, J. (2015) Public transport demand, in: C. Nash (Eds) *Handbook of Research Methods and Applications in Transport Economics and Policy.*, Edward Elgar, 192-211.

Preston, J., Wall, G., Batley, R., Ibáñez, J. and Shires, J. (2009) Impact of Delays on Passenger Train Services: Evidence from Great Britain. *Transportation Research Record*, 2117, 14-23.

Pröger, T. (2016) *Robust Routing in Urban Public Transportation Networks*. PhD Thesis: ETH Zurich.

Pyrga, E., Schulz, F., Wagner, D. and Zaroliagis, C. (2004) Experimental comparison of shortest path approaches for timetable information. In *Proceedings of the 6th Workshop on Algorithm Engineering and Experiments (ALENEX'04)*, 88-99.

Pyrga, E., Schulz, F., Wagner, D. and Zaroliagis, C. (2008) Efficient models for timetable information in public transportation systems. *ACM Journal of Experimental Algorithmics*, 12(2.4): 1-39.

Rietveld, P. (2005) Six reasons why supply-oriented indicators systematically overestimate service quality in public transport. *Transport Reviews*, 25(3), 319-328.

Rietveld, P., Bruinsma, F.R. and van Vuuren, D.J. (2001) Coping with unreliability in public transport chains: A case study for Netherlands. *Transportation Research Part A*, 35, 539-559.

RRUKA (2015) *Data to Improve Customer Experience*. Competition launch workshop. The Rail Research UK Association. http://live.wavecast.co/data-to-improve-the-customer-experience/programme/ [Accessed 20 November 2016]

Schnee, M. (2009) *Fully Realistic Multi-Criteria Timetable Information Systems*. PhD Thesis, Technische Universität Darmstadt.

Schulz, F., Wagner, D., and Weihe, K. (2000) Dijkstra's algorithm on-line: An empirical case study from public railroad transport. *ACM Journal of Experimental Algorithmics*, 5(12), 1-23.

Senbil, M. and Kitamura, R. (2004) Heterogeneity in commuter departure time decision: a prospect theoretic approach, in: N.O. Attoh-Okine and B. Ayyub (Eds) *Applied Research in Uncertainty Modelling and Analysis*, pp. 369-398.

Small, K.A. (1982) The scheduling of consumer activities: work trips. *American Economic Review*, 72, 172-181.

UIC (International Union of Railways) (2000) *Timetable recovery margins to guarantee timekeeping - recovery margins*. Leaflet 451-1. Paris, France.

Vromans, M. (2005) *Reliability of Railway Systems*. PhD Thesis: Erasmus University Rotterdam.

Wagner, D. (2015) *Route Planning Algorithms in Transportation Networks*. Presentation at 7th International Network Optimization Conference, Warsaw, May 18 2015.

Watson, R. (2008) *Train planning in a fragmented railway – a British perspective*. PhD Thesis, Loughborough University.

Weston, P., Goodman, C., Takagi, R., Bouch, C., Armstrong, J., and Preston, J. (2006) *Minimising train delays in a mixed-traffic railway network with consideration of passenger delay*. 7th World Congress on Railway Research, Montréal, Canada.

Yuan, J. (2006) *Stochastic Modelling of Train Delays and Delay Propagation in Stations*. PhD Thesis, Delft University of Technology.

Yuan, J. and Medeossi, G. (2014) Statistical Analysis of Train Delays and Movements, in: I.A. Hansen and J. Pachl (Eds) *Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation.*, pp. 217-236.

Appendix A

A survey of current pre-trip journey planning systems

Table A1 A survey of current pre-trip journey planning systems (Surveyed in early April, 2018)

Country (A to Z)	Main information provider	Line schedules (static)	Transfer-involved journey planning (static)	Customisable transfer times	Frequently updating the underlying timetables
Argentina	SOFSE	Y	N	N	N
Australia	regional operators e.g. NSW TrainLink and V/Line	Y	Y	N	N
Austria	ÖBB	Y	Y	Y	Y
Bangladesh	Bangladesh Railway	Y	N	N	N
Belarus	Belarusian Railway (BCh)	Y	Y	N	N
Belgium	NMBS/SNCB	Y	Y	Y	Y
Brazil	commuter rail operators e.g. CPTM and SuperVia	Y	Y	N	N
Canada	VIA Rail and several commuter rail operators	Y	Y	N	N
China	CR	Y	N	N	N
Croatia	HŽPP	Y	Y	N	N
Czech Republic	ČD	Y	Y	Y	Y
Denmark	DSB	Y	Y	N	N
Egypt	ENR	Y	N	N	N
Finland	VR	Y	Y	N	N
France	SNCF	Y	Y	N	N
Germany	DB Bahn	Y	Y	Y	Y
Hungary	MÁV	Y	Y	Y	Y
India	IR	Y	N	N	N
Indonesia	Persero	Y	N	N	N
Iran	RAI	Y	N	N	N
Israel	Israel Railways	Y	Y	N	N
Italy	Trenitalia and Italo NTV	Y	Y	N	N
Japan	JR Group	Y	Y	N	N
Kazakhstan	KTZ	Y	N	N	N

Luxemburg	CFL	Y	Y	N	N
Malaysia	KTMB	Y	N	N	N
Morocco	ONCF	Y	N	N	N
Netherlands	NS	Y	Y	Y	Y
Norway	NSB	Y	Y	N	N
Pakistan	PR	Y	N	N	N
Poland	PKP Group	Y	Y	N	N
Portugal	CP and several commuter rail operators	Y	Y	N	N
Romania	CFR	Y	Y	N	N
Russia	OAO	Y	Y	N	N
South Africa	Shosholoza Meyl	Y	N	N	N
South Korea	Korail	Y	Y	N	N
Spain	Renfe	Y	Y	N	N
Sweden	SJ	Y	Y	N	N
Switzerland	SBB, BLS, etc.	Y	Y	Y	Y
Taiwan	THSR	Y	N	N	N
Thailand	SRT	Y	N	N	N
Turkey	TCDD	Y	N	N	N
Ukraine	Ukrainian Railways	Y	Y	N	N
United Kingdom	RDG/National Rail	Y	Y	Y	Y
United States	Amtrack and various commuter rail operators	Y	Y	N	N

Appendix B

A Python implementation of CRF

Note: the original source codes are a large collection of many functionally independent sub-procedures, each of which is separately composed and stored (in separate files). Here, in order to avoid messiness, these sub-procedures are grouped (by general functionality) into three files named data-preprocessing.py, screening-1.py, and screening-2.py, respectively. Generally, the file named data-preprocessing.py can be thought of as the implementation of Steps 1 and 2 in CRF, screening-1.py the implementation of Step 3, and screening-2.py the implementation of Steps 4 and 5. Within each file, however, the sub-procedures are not necessarily organised in the same order with the pseudo code presented in Subsection 3.6.3. This is because there exists no particular priority between certain sub-procedures, and it makes no difference to execute one before another.

(1) data-preprocessing.py

```python
import pandas as pd
from datetime import datetime
import re
from ord_set import OrderedSet
from time import time
from math import sin, cos, sqrt, atan2, radians


## author: Yiwei Guo   02/05/2016 @soton


regex = re.compile('\s+to\s+|\s+\([A-Z]{2}\)')

data = pd.read_csv('routes.txt')

temp = list(data['route_long_name'].values)

L1 = [regex.split(each) for each in temp]

L2 = [each[0] for each in L1]

L1 = [each[1] for each in L1]

data['ori'] = L2
data['des'] = L1

#data = data[data['route_type'] == 2]

data.to_csv('route-2.csv', index=False)

data1 = pd.read_csv('stop-3.csv', usecols = ['stop_name', 'stop_lat', 'stop_lon'])

data = pd.merge(data, data1, left_on='des', right_on='stop_name')

#data = data.sort_values(by='route_id')

data.to_csv('route-3.csv', index=False)




data = pd.read_csv('stop-2.txt', usecols = ['stop_name', 'parent_station', 'stop_lat',
'stop_lon'])

data = data.drop_duplicates(['stop_name'])

data.to_csv('stop-3.csv', index=False)




data = pd.read_csv('calendar.txt')

delta = datetime.strptime('23:59', '%H:%M').time()

start_date = [str(x) for x in data['start_date']]

end_date = [str(x) for x in data['end_date']]

start_date = [datetime.strptime(x, '%Y%m%d') for x in start_date]

end_date = [datetime.strptime(x, '%Y%m%d').date() for x in end_date]

end_date = [datetime.combine(x, delta) for x in end_date]

data['s_date'] = start_date
data['e_date'] = end_date
```

```python

data = data.assign(diff = lambda x: x['e_date'] - x['s_date'])

data1 = data['diff'].astype('timedelta64[D]')

data['diff2'] = data1

del data['start_date']
del data['end_date']
del data['diff']

#data = data.reindex(columns = ['service_id', 'saturday', 'sunday', 's_date', 'e_date', 'diff2'])

data.to_csv('calendar-2.csv', index=False)




'''
data = pd.read_csv('calendar-2.csv')

s1 = pd.to_datetime(data['s_date'])
s2 = pd.to_datetime(data['e_date'])

data['sd'] = s1
data['ed'] = s2

dep_day = datetime.strptime(raw_input('Departure Date (DD/MM/YY): '), '%d/%m/%y')
#25/01/17

day_of_week = datetime.strftime(dep_day, '%A').lower()

data1 = data[data[day_of_week] == 1]

data = data1[(data1['sd'] <= dep_day) & (data1['ed'] >= dep_day)]

L1 = list(data['service_id'].values)

#data.to_csv('query-calendar.csv',index=False)

data = pd.read_csv('trips.txt', usecols=['trip_id', 'service_id', 'route_id'])

#data1 = pd.read_csv('query-calendar.csv', usecols=['service_id'])

mask = data['service_id'].isin(L1)

data = data[mask]

del data['service_id']

data.to_csv('trip-5.csv',index=False)

L1 = list(data['trip_id'].values)

data = pd.read_csv('stimes-2.csv', usecols=['trip_id', 'parent_station', 'stop_sequence'])

mask = data['trip_id'].isin(L1)

data = data[mask]

data = data.sort_values(by=['trip_id', 'stop_sequence'])

data.to_csv('query-sch-1.csv', index=False)

'''
```

```python
data1 = pd.read_csv('trip-5.csv')

data = pd.read_csv('incoming-1.csv',  header=None)

L1 = list(data[0].values)

L2 = []
for each in L1:
    data3 = data1[data1['route_id']==each]
    ind = data3.index
    temp = ind[len(data3)/2]
    L2.append(data3.at[temp, 'trip_id'])


data = pd.read_csv('outgoing-1.csv',  header=None)

L1 = list(data[0].values)

L3 = []
for each in L1:
    data3 = data1[data1['route_id']==each]
    ind = data3.index
    temp = ind[len(data3)/2]
    L3.append(data3.at[temp, 'trip_id'])


data = pd.read_csv('query-sch-1.csv', usecols=['parent_station', 'trip_id'])

L4 = []
for each in L2:
    data3 = data[data['trip_id']==each]
    temp = tuple(data3['parent_station'].values)
    L4.append(temp)

L5 = []
for each in L3:
    data3 = data[data['trip_id']==each]
    temp = tuple(data3['parent_station'].values)
    L5.append(temp)

ee = [OrderedSet(each) for each in L4]

dd = [OrderedSet(each) for each in L5]

L4 = pd.Series(L4)
L4.to_csv('incoming-stops.csv', index=False)

L5 = pd.Series(L5)
L5.to_csv('outgoing-stops.csv', index=False)

ff = OrderedSet()

start1 = time()
for each in ee:
    for every in dd:
        if each[0] == every[-1] and each[-1] == every[0]:
            continue
        if each[-1] == every[-1] or each[0] == every[0]:
            continue
        temp3 = each & every
        if len(temp3) == 1:
            ff.add(temp3[0])

end1 = time()
```

```
print round(end1-start1, 6)
ee = pd.Series(list(ff))
ee.to_csv('transtations.csv', index=False)



'''
data = pd.read_csv('stop_times.txt', usecols = ['trip_id', 'stop_id', 'stop_sequence', 'arrival_time', 'departure_time', 'platform'])

arrival_time = list(data['arrival_time'].values)
departure_time = list(data['departure_time'].values)
arrival = []
departure = []

for each in arrival_time:
    hour = int(each[:2])
    minute = int(each[3:5])
    converted = hour*60 + minute
    arrival.append(converted)

for each in departure_time:
    hour = int(each[:2])
    minute = int(each[3:5])
    converted = hour*60 + minute
    departure.append(converted)

data['arrival'] = arrival
data['departure'] = departure

data1 = pd.read_csv('stop-2.txt',usecols=['stop_id', 'parent_station'])
data = pd.merge(data, data1, on='stop_id')

#data.to_csv('stimes-1.csv', index=False)

del data['arrival_time']
del data['departure_time']
del data['stop_id']

data = data.reindex(columns=['trip_id', 'parent_station', 'stop_sequence', 'platform', 'arrival', 'departure'])
data.to_csv('stimes-2.csv', index=False)
'''




start1 = time()

L1 = [line.strip() for line in open('transtations.csv')]
L1 = set(L1)
L2 = {'BFR', 'CST', 'CTK', 'CHX', 'EUS', 'FST', 'KGX', 'KPA', 'LST', 'LBG', 'MYB', 'PAD', 'STP', 'VIC', 'WAT'}

L1 = list(L1 - L2)

data = pd.read_csv('stimes-2.csv', dtype={'platform':object})

data = data[(data['arrival'] > 479) & (data['departure'] < 1021)]

L2 = data['parent_station'].isin(L1)

data = data[L2]

data1 = pd.read_csv('trips.txt', usecols = ['trip_id', 'route_id', 'service_id'])
data2 = pd.read_csv('route-3.csv', usecols = ['route_id', 'route_type', 'ori', 'des', 'stop_lat', 'stop_lon'])
data3 = pd.read_csv('stop-3.csv', usecols = ['parent_station', 'stop_lat', 'stop_lon'])
```

```
data = pd.merge(data,data1,how='inner',on='trip_id')
data = pd.merge(data,data3,how='inner',on='parent_station')
data = pd.merge(data,data2,how='inner',on='route_id')

data = data[data['route_type']==2]

#del data['des']
del data['route_type']
del data['trip_id']
#data.to_csv('intermediate-1.csv', index=False)

R = 6371.0

a1 = data['stop_lat_x']
b1 = data['stop_lon_x']
a2 = data['stop_lat_y']
b2 = data['stop_lon_y']

del data['stop_lat_x']
del data['stop_lat_y']
del data['stop_lon_x']
del data['stop_lon_y']


l1 = data.index
dist = []

for i in l1:
    lat1 = radians(a1[i])
    lon1 = radians(b1[i])
    lat2 = radians(a2[i])
    lon2 = radians(b2[i])
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    dlon = (sin(dlat/2))**2 + cos(lat1) * cos(lat2) * (sin(dlon/2))**2
    dlat = 2 * atan2(sqrt(dlon), sqrt(1-dlon))
    dlon = R * dlat
    dist.append(dlon)

data['dist'] = dist


data2 = pd.read_csv('calendar-2.csv', usecols = ['service_id', 'saturday', 'sunday', 'diff2'])

data = pd.merge(data,data2,on='service_id')

data = data[data.diff2 > 59]

data = data.sort_values(by='arrival')

end1 = time()
print round(end1-start1, 6)

data2 = data[(data.saturday != 1) & (data.sunday != 1)]

data2.to_csv('stimes-wday.csv', index=False)

data2 = data[data.saturday == 1]

data2.to_csv('stimes-sat.csv', index=False)

data = data[data.sunday == 1]

data.to_csv('stimes-sun.csv', index=False)
```

```python


data = pd.read_csv('trip-5.csv')

data1 = pd.read_csv('larthan90days.csv', header=None)

L1 = list(data1[0].values)

mask = data['route_id'].isin(L1)

data1 = data[mask]

data1 = data1['route_id'].drop_duplicates()

data1.to_csv('incoming-1.csv', index=False)

data1 = pd.read_csv('conn-routes.csv', header=None)

L1 = list(data1[0].values)

mask = data['route_id'].isin(L1)

data1 = data[mask]

data1 = data1['route_id'].drop_duplicates()

data1.to_csv('outgoing-1.csv', index=False)




'''
data = pd.read_csv('routes.txt', usecols=['route_id', 'route_long_name'])

regex = re.compile('\s+to\s+|\s+\([A-Z]{2}\)')

temp = list(data['route_long_name'].values)

L1 = [regex.split(each) for each in temp]

L2 = [each[0] for each in L1]

L1 = [each[1] for each in L1]

data['ori'] = L2
data['des'] = L1


data1 = pd.read_csv('stop-3.csv', usecols = ['stop_name', 'stop_lat', 'stop_lon'])

_ = data1.rename(columns={'stop_name':'ori', 'stop_lat':'lat_1', 'stop_lon':'lon_1'},
inplace=True)

data = pd.merge(data, data1, on='ori')

_ = data1.rename(columns={'ori':'des', 'lat_1':'lat_2', 'lon_1':'lon_2'}, inplace=True)

data = pd.merge(data, data1, on='des')

data.to_csv('route_4.csv', index=False)

dist = []
R = 6371.0

for i in range(len(data)):
    lat1 = radians(data.at[i, 'lat_1'])
    lon1 = radians(data.at[i, 'lon_1'])
```

```
    lat2 = radians(data.at[i, 'lat_2'])
    lon2 = radians(data.at[i, 'lon_2'])
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    dlon = (sin(dlat/2))**2 + cos(lat1) * cos(lat2) * (sin(dlon/2))**2
    dlat = 2 * atan2(sqrt(dlon), sqrt(1-dlon))
    dlon = R * dlat
    dist.append(dlon)

data['dist'] = dist

data.to_csv('route_5.csv', index=False)

data = data[data['dist']>39]

data['route_id'].to_csv('larthan40.csv', index=False)

L1 = list(data['route_id'].values)


data = pd.read_csv('calendar-2.csv', usecols=['service_id', 'diff2'])

data = data[data['diff2']>90]

data1 = pd.read_csv('trips.txt', usecols=['route_id', 'service_id', 'trip_id'])

data1 = pd.merge(data1, data, on='service_id')

data1['trip_id'].to_csv('trip-6.csv', index=True)

L2 = data1['route_id'].drop_duplicates()

L2.to_csv('larthan90days.csv', index=False)

L2 = list(L2.values)

L1 = pd.Series(L1)

mask = L1.isin(L2)

L1 = L1[mask]

L1.to_csv('conn-routes.csv', index=False)

'''



data = pd.read_csv('transfers.txt', usecols=['from_stop_id', 'min_transfer_time'])

data = data.assign(ch2 = lambda x: x['min_transfer_time']/60)

data['ch2'] = data['ch2'].astype('int32')

del data['min_transfer_time']

data.to_csv('transfers-sou.csv', index=False)


#data = pd.read_csv('transfers-sou.csv')

high = []

for each in data['ch2']:
    if each <= 5:
        high.append(each + 4)
    elif 5 < each < 8:
        high.append(each + 2)
    else:
        high.append(each + 1)

data['high'] = high

data.to_csv('scan-bounds.csv', index=False)
```

(2) screening-1.py

```python
import pandas as pd
import os
from time import time
from bisect import bisect_left


## author: Yiwei Guo   02/05/2016  @soton


bounds = pd.read_csv('scan-bounds.csv', index_col='from_stop_id')

os.chdir('20161122')
#os.chdir('temp')

cdir = os.getcwd()
L1 = os.listdir(cdir)
candidate = []
start1 = time()

for fname in L1:
    data = pd.read_csv(fname, dtype={'platform':object})
    [lo, hi] = [bounds.loc[fname[0:3], 'ch2'], bounds.loc[fname[0:3], 'high']]
    if lo > 11:
        continue

    r_list = data['route_id'].drop_duplicates()
    r_list = list(r_list.values)
    list_conn = []

    for each in r_list:
        data1 = data[data['route_id']==each]
        L1 = len(data1) - 1
        diff1 = []
        for i in range(L1):
            diff = data1.iat[i+1,4] - data1.iat[i,4]
            diff1.append(diff)
        if diff1 != []:
            if max(diff1)>60:
                list_conn.append(each)

    if len(list_conn) == 0:
        continue
    data1 = data[data['stop_sequence'] != 1]
    if data1.empty:
        continue
    r_list = data1['route_id'].drop_duplicates()
    list_fdr = list(r_list.values)

    result = []
    L1 = len(data) - 1

    for i in range(L1):
        if data.iat[i,5] in list_fdr:
            flag = data.iat[i,3]
            flag1 = data.iat[i,5]
            data1 = data[(data['departure'] >= flag + lo) & (data['departure'] <= flag
            + hi)]
            if data1.empty:
                continue
            data1 = data1[data1['dist'] >= 40]
            if data1.empty:
                continue
            data1 = data1[data1['route_id'] != flag1]
            if data1.empty:
                continue
            flag = data1['route_id'].isin(list_conn)
            data1 = data1[flag]
```

```
            if data1.empty:
                continue
            flag1 = len(data1)
            for j in range(flag1):
                flag = [data.iat[0,0], data.iat[i,3], data1.iat[j,4], data.iat[i,2],
                data1.iat[j,2], data.iat[i,5], data1.iat[j,5], data.iat[i,6],
                data1.iat[j,6], data.iat[i,7], data.iat[i,8], data1.iat[j,7],
                data1.iat[j,8], data1.iat[j,9]]
                candidate.append(flag)

end1 = time()
print round(end1-start1, 6)

if len(candidate) == 0:
    print 'No Critical Connections found! Empty candidate list!'
else:
    candidate = pd.DataFrame(candidate, columns=['trans', 'arr_f', 'dep_c', 'pl_f',
    'pl_c', 'ru_f', 'ru_c', 'serv_f', 'serv_c', 'ori_f', 'des_f', 'ori_c', 'des_c',
    'c_dist'])

    candidate = candidate[candidate['serv_f']==candidate['serv_c']]
    flag = (candidate['ori_f']!=candidate['ori_c']) &
    (candidate['des_f']!=candidate['des_c']) & (candidate['ori_f']!=candidate['des_c'])
    & (candidate['ori_f']!=candidate['des_f']) & (candidate['ori_c']!=candidate['des_c'])
    candidate = candidate[flag]

    if candidate.empty:
        print 'No Critical Connections found! No service dates and routes satisfied!'
    else:
        candidate = candidate[candidate['pl_f']!=candidate['pl_c']]
        if candidate.empty:
            print 'No Critical Connections found! No platforms satisfied!'
        else:
            candidate = candidate.assign(r_con = lambda x: x['ori_f'] + x['trans'] +
            x['des_c'])
            hi = candidate['r_con'].value_counts()
            flag1 = list(hi.index)
            lo = list(hi.values)
            hi = pd.DataFrame({'r_con':flag1, 'p_counts':lo})
            candidate = pd.merge(candidate, hi, on='r_con')
            candidate = candidate.drop_duplicates(['r_con'])
            candidate = candidate[(candidate['p_counts'] >= 4) & (candidate['p_counts']
            <= 9)]
            candidate.to_csv('critical-routes.csv', index=False)




conns = [line.strip().split(",") for line in open('sch-lar-2.csv') if not
line.startswith('Sd') ]

conns = [int(each[2]) for each in conns]

L1 = range(1, 1440)

L2 = [bisect_left(conns, each) for each in L1]

L2 = pd.Series(L2)

L2.to_csv('begin-points-lar.csv', index=True)


'''
data1 = pd.read_csv('sch-lar-2.csv')

data1 = data1[data1['ruid']==4901]
```

```
data1 = data1.sort_values(by=['tpid', 'dt'])

data1.to_csv('4901.csv', index=False) '''



data1 = pd.read_csv('sch-lar.csv')

data3 = data1.copy()

data3 = data3.drop_duplicates(['tpid'])

data3 = data3[['Sd', 'dt', 'tpid']]

_ = data3.rename(columns={'Sd':'ors', 'dt':'odt'}, inplace=True)

data3 = data3.reindex(columns=['tpid', 'ors', 'odt'])

data1 = pd.merge(data1, data3, on='tpid')

data1 = data1.sort_values(by='dt')

data1.to_csv('sch-lar-2.csv', index=False)

data1 = pd.read_csv('sch-2.csv')

data3 = data1.copy()

data3 = data3.drop_duplicates(['tpid'])

data3 = data3[['Sd', 'dt', 'tpid']]

_ = data3.rename(columns={'Sd':'ors', 'dt':'odt'}, inplace=True)

data3 = data3.reindex(columns=['tpid', 'ors', 'odt'])

data1 = pd.merge(data1, data3, on='tpid')

data1 = data1.sort_values(by='dt')

data1.to_csv('sch-22.csv', index=False)



data1 = pd.read_csv('trip-5.csv')

data2 = pd.read_csv('lul-sch-1.csv', usecols=['tpid', 'ruid'])

data2 = data2.drop_duplicates(['tpid'])

data2 = data2.reindex(columns=['ruid', 'tpid'])

_ = data2.rename(columns={'ruid':'route_id', 'tpid':'trip_id'}, inplace=True)

data1 = pd.concat([data1, data2])

data1.to_csv('trip-lar.csv', index=False)



data = pd.read_csv('stop-3.csv')

data = data.drop_duplicates(['parent_station'])

data2 = pd.read_csv('stop-dicts.txt', usecols=['parent_station', 's_num'])
```

```python
L2 = data[data['stop_lat'] > 52.06]

L2 = pd.merge(L2, data2, how='inner', on='parent_station')

L2 = L2[['parent_station', 's_num']]

L2.to_csv('northen-stops.csv', index=False)

m1 = data['stop_lat'] < 51.283
m2 = (data['stop_lon'] < -0.512) & (data['stop_lat'] < 51.69)
m3 = (data['stop_lon'] > 0.332) & (data['stop_lat'] < 51.69)

m4 = m1 | m2 | m3

data = data[m4]

data = pd.merge(data, data2, how='inner', on='parent_station')

data = data[['parent_station', 's_num']]

data.to_csv('southern-stops.csv', index=False)




data1 = pd.read_csv('sch-2.csv')

data2 = pd.read_csv('lul-sch-1.csv', usecols=['Sd2', 'Sa2', 'dt', 'at', 'tpid', 'ruid'])

_ = data2.rename(columns={'Sd2':'Sd', 'Sa2':'Sa'}, inplace=True)

data1 = pd.concat([data1, data2])

data1 = data1.sort_values(by='dt')

data1.to_csv('sch-lar.csv', index=False)



L3 = []

for x in range(1440):
    if x < L1[0]:
        L3.append(0)
    elif x >= L1[-1]:
        L3.append(L2[-1])
    else:
        for i in range(len(L1)-1):
            if L1[i] <= x < L1[i+1]:
                L3.append(L2[i])

L3 = pd.Series(L3)




data = pd.read_csv('stop-crs.csv', usecols=['stop_id', 'stop_name'])

temp = range(len(data))

data['stop_num'] = temp

_ = data.rename(columns={'stop_id':'parent_station', 'stop_name':'s_name', 'stop_num':'s_num'}, inplace=True)

data.to_csv('stop-dicts.txt', index=False)
```

```
data1 = pd.read_csv('trip-5.csv')

data2 = pd.read_csv('trips.txt', usecols=['trip_id', 'atoc_code'])

data1 = pd.merge(data1,data2,on='trip_id')

data1.to_csv('trip-7.csv', index=False)



data1 = pd.read_csv('stimes-2.csv', usecols=['trip_id', 'parent_station',
'stop_sequence', 'arrival', 'departure'])

data2 = pd.read_csv('trip-7.csv')

data2 = data2[data2['atoc_code']!='LT']
#mask = data2['atoc_code'].isin(['LT', 'LO', 'XR'])
#mask2 = -mask
#data2 = data2[mask2]

data1 = pd.merge(data1, data2, how='inner', on='trip_id')

data3 = pd.read_csv('stop-dicts.txt', usecols=['parent_station', 's_num'])

data1 = pd.merge(data1, data3, how='inner', on='parent_station')

del data1['atoc_code']

del data1['parent_station']

data1 = data1.sort_values(by=['trip_id','stop_sequence'])

data1.to_csv('sch-1.csv', index=False)



data1 = pd.read_csv('sch-1.csv')

result = []

for i in range(len(data1)-1):
    if data1.iat[i, 1] >= data1.iat[i+1, 1]:
        continue
    else:
        hit = [data1.iat[i, 5], data1.iat[i+1, 5], data1.iat[i, 3], data1.iat[i+1, 2],
        data1.iat[i, 0], data1.iat[i, 4]]
        result.append(hit)

result = pd.DataFrame(result, columns=['Sd', 'Sa', 'dt', 'at', 'tpid', 'ruid'])

result = result.sort_values(by = 'dt')

result.to_csv('sch-2.csv', index=False)




data1 = pd.read_csv('lul-11.csv')
data2 = pd.read_csv('lul-21.csv')
idxs =
['BFR','CST','CTK','CHX','EUS','FST','KGX','KPA','LST','LBG','MYB','PAD','STP','VIC','WAT
']
idxs2 = {id:v for id,v in enumerate(idxs)}
_ = data1.rename(index=idxs2, inplace=True)
```

```
_ = data2.rename(index=idxs2, inplace=True)

L1 = []

for i in idxs:
    for j in idxs:
        if data1.at[i,j] != 0:
            if data2.at[i,j] == 'Walking' or data2.at[i,j] == 'Walk':
                temp = [i, j, data1.at[i,j], data2.at[i,j], 1]
            else:
                temp = [i, j, data1.at[i,j], data2.at[i,j], 3]
            L1.append(temp)

# S1 = [each[0] for each in L1]
# S2 = [each[1] for each in L1]
# T1 = [int(each[2]) for each in L1]
# D1 = [str(each[3]) for each in L1]
# F1 = [int(each[4]) for each in L1]

#data = {'from': S1, 'to': S2, 'time': T1, 'lines': D1, 'freq': F1}
L1 = pd.DataFrame(L1, columns=['from', 'to', 'time', 'lines', 'freq'])
L1.to_csv('lul-41.csv', index=False)


data = pd.read_csv('lul-41.csv')
trip_id = 700000
L1 = []

for i in range(len(data)):
    s1 = data.iloc[i][0]
    s2 = data.iloc[i][1]
    t1 = data.iloc[i][2]

    for j in range(360,1410,data.iloc[i][4]):
        temp = [s1, s2, j, j+t1, trip_id]
        L1.append(temp)

    trip_id += 1

L1 = pd.DataFrame(L1, columns=['Sd', 'Sa', 'dt', 'at', 'tpid'])
L1.to_csv('lul-sch-0.csv', index=False)




data = pd.read_csv('lul-sch-0.csv')

data2 = pd.read_csv('stop-dicts.txt', usecols=['parent_station', 's_num'])

_ = data2.rename(columns={'parent_station':'Sd', 's_num':'Sd2'}, inplace=True)

data = pd.merge(data, data2, on='Sd')

_ = data2.rename(columns={'Sd':'Sa', 'Sd2':'Sa2'}, inplace=True)

data = pd.merge(data, data2, on='Sa')

del data['Sd']
del data['Sa']

data['ruid'] = 0

data['r_name'] = 'London Underground'

data = data.reindex(columns=['Sd2', 'Sa2', 'dt', 'at', 'tpid', 'ruid', 'r_name'])

data.to_csv('lul-sch-1.csv', index=False)
```

```python
data2 = pd.read_csv('scan-bounds.csv', usecols=['from_stop_id', 'low'])

_ = data2.rename(columns={'from_stop_id':'parent_station', 'low':'ch1'}, inplace=True)

data1 = pd.merge(data1, data2, how='inner', on='parent_station')



'''
data3 = pd.read_csv('transfers.txt', usecols=['from_stop_id', 'min_transfer_time'])

#data3 = data3.assign(ch2=lambda x: x['min_transfer_time']/60)
temp = [each/60 for each in data3['min_transfer_time']]

data3['ch2'] = temp

del data3['min_transfer_time']

data3.to_csv('transfers-2.csv', index=False)  '''
```

(3) screening-2.py

```python
import pandas as pd
from math import sin, cos, sqrt, atan2, radians
from array import array
from time import time
from ord_set import OrderedSet
#from random import sample


## author: Yiwei Guo  02/05/2016 @soton


data = pd.read_csv('critical-routes.csv', usecols=['trans', 'pl_f', 'pl_c', 'ru_f',
'ru_c', 'serv_f', 'c_dist', 'ori_f', 'des_f', 'ori_c', 'des_c', 'p_counts'],
dtype={'pl_f': object, 'pl_c': object})

data2 = pd.read_csv('stop-3.csv', usecols=['stop_name', 'stop_lat', 'stop_lon'])

_ = data2.rename(columns={'stop_name':'ori_f', 'stop_lat':'lat_of',
'stop_lon':'lon_of'}, inplace=True)

data = pd.merge(data, data2, on='ori_f')

_ = data2.rename(columns={'ori_f':'des_c', 'lat_of':'lat_dc', 'lon_of':'lon_dc'},
inplace=True)

data = pd.merge(data, data2, on='des_c')

data2 = pd.read_csv('stop-3.csv', header=0, names=['stops', 'lat_df', 'lon_df',
'trans'], usecols=['trans', 'lat_df', 'lon_df'])

data = pd.merge(data, data2, on='trans')

R = 6371.0

a1 = data['lat_of']
b1 = data['lon_of']
a2 = data['lat_dc']
b2 = data['lon_dc']

del data['lat_dc']
del data['lon_dc']
del data['lat_of']
del data['lon_of']

l1 = data.index
dist = []

for i in l1:
    lat1 = radians(a1[i])
    lon1 = radians(b1[i])
    lat2 = radians(a2[i])
    lon2 = radians(b2[i])
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    dlon = (sin(dlat/2))**2 + cos(lat1) * cos(lat2) * (sin(dlon/2))**2
    dlat = 2 * atan2(sqrt(dlon), sqrt(1-dlon))
    dlon = R * dlat
    dist.append(dlon)

data['r_dist'] = dist

'''
a2 = data['lat_df']
b2 = data['lon_df']

del data['lat_of']
del data['lon_of']
```

```
del data['lat_df']
del data['lon_df']

dist = []

for i in l1:
    lat1 = radians(a1[i])
    lon1 = radians(b1[i])
    lat2 = radians(a2[i])
    lon2 = radians(b2[i])
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    dlon = (sin(dlat/2))**2 + cos(lat1) * cos(lat2) * (sin(dlon/2))**2
    dlat = 2 * atan2(sqrt(dlon), sqrt(1-dlon))
    dlon = R * dlat
    dist.append(dlon)

data['f_dist'] = dist  '''

data = data[data['r_dist'] >= 40]


'''
data.to_csv('intermediate.csv', index=False)

a1 = (data['r_dist'] >= data['c_dist']) & (data['r_dist'] >= data['f_dist'])

a2 = (data['r_dist'] > data['f_dist']) & (data['r_dist'] < data['c_dist']) & (data['f_dist'] + data['c_dist'] <= 2*data['r_dist'])

b1 = (data['r_dist'] < data['f_dist']) & (data['r_dist'] > data['c_dist']) & (data['f_dist'] + data['c_dist'] <= 1.4*data['r_dist'])

b2 = a1 | b1 | a2

data = data[b2] '''

_ = data.fillna({'pl_f':'0', 'pl_c':'0'}, inplace=True)

a2 = []
for each in data['pl_f']:
    if len(each) == 1:
        a2.append(int(each))
    elif len(each) == 3:
        a2.append(int(each[0:-1]))
    elif ord(each[-1]) > 64:
        a2.append(int(each[0:-1]))
    else:
        a2.append(int(each))


b2 = []
for each in data['pl_c']:
    if len(each) == 1:
        b2.append(int(each))
    elif len(each) == 3:
        b2.append(int(each[0:-1]))
    elif ord(each[-1]) > 64:
        b2.append(int(each[0:-1]))
    else:
        b2.append(int(each))


data['p_f'] = a2
data['p_c'] = b2

del data['pl_f']
```

```
del data['pl_c']

data = data.assign(pl_diff=lambda x: abs(x['p_c']-x['p_f']))

data2 = pd.read_csv('scan-bounds.csv', header=0, names=['trans', 'lo', 'hi'],
usecols=['trans', 'lo'])

#data2 = pd.read csv('scan-bounds.csv', usecols=['from stop id', 'low'])

data = pd.merge(data, data2, on='trans')

a1 = (data['lo'] <= 5) & (data['pl_diff'] > 0)
b1 = (data['lo'] > 5) & (data['lo'] <= 7) & (data['pl_diff'] > 1)
a2 = (data['lo'] > 7) & (data['pl diff'] > 2)

b2 = a1 | b1 | a2

data = data[b2]

#data.to_csv('critical-routes-2.csv', index=False)
data.to_csv('critical-routes-3.csv', index=False)



#945
#0.526673


MAX_STATIONS = 2650
MAX_INT = 1000000
MAX INT2 = 3000

class Connection:
    def   init  (self, line):
        tokens = line.split(",")
        self.departure_station = int(tokens[0])
        self.arrival_station = int(tokens[1])
        self.departure_timestamp = int(tokens[2])
        self.arrival_timestamp = int(tokens[3])
        self.trip_id = int(tokens[4])
        self.route_id = int(tokens[5])
        self.ori station = int(tokens[6])
        self.ori_dt = int(tokens[7])



class Timetable:

    def __init__(self, filename):
        self.connections = [Connection(line.strip()) for line in open(filename) if not
        line.startswith('Sd')]


class CSA:
    def __init__(self):
        self.timetable0 = Timetable('sch-lar-2.csv')
        self.timetable1 = Timetable('sch-22.csv')
        self.in_connection = array('I')
        self.earliest arrival = array('H')
        self.trip_flag = {}
        #self.counters = 0
        self.results = []
        temp = [line.strip().split(',') for line in open('transfers-sou.csv') if not
        line.startswith('from_stop_id')]
        self.ch time = [int(each[1]) for each in temp]
        temp = [line.strip().split(',') for line in open('trip-lar.csv') if not
        line.startswith('route_id')]
```

```
        self.trips = [int(each[1]) for each in temp]
        temp = [line.strip().split(',') for line in open('begin-points-lar.csv')]
        self.bpl =  array('I', [int(each[1]) for each in temp])
        temp = [line.strip().split(',') for line in open('begin-points.csv')]
        self.bps =  array('I', [int(each[1]) for each in temp])
        temp = [line.strip().split(',') for line in open('northen-stops.csv') if not
        line.startswith('parent_station')]
        self.norths = [int(each[1]) for each in temp]
        temp = [line.strip().split(',') for line in open('southern-stops.csv') if not
        line.startswith('parent_station')]
        self.souths = [int(each[1]) for each in temp]

        #temp = [line.strip().split(',') for line in open('stop-dicts.txt') if not
        line.startswith('parent station')]
        #self.dict3 = {each[0]:int(each[2]) for each in temp}


    def main_loop(self, conns, arrival_station):
        earliest = MAX_INT

        for i, c in enumerate(conns):
            if c.departure_timestamp > earliest:
                break
            if self.trip_flag[c.trip_id] == 1:
                if c.departure_timestamp >= self.earliest_arrival[c.departure_station]
                and c.arrival_timestamp < self.earliest_arrival[c.arrival_station]:
                    self.earliest_arrival[c.arrival_station] = c.arrival_timestamp
                    self.in_connection[c.arrival_station] = i
                    if c.arrival_station == arrival_station:
                        earliest = min(earliest, c.arrival_timestamp)
            else:
                if c.departure_timestamp >= self.earliest_arrival[c.departure_station]
                + self.ch_time[c.departure_station] and c.arrival_timestamp <
                self.earliest_arrival[c.arrival_station]:
                    self.earliest_arrival[c.arrival_station] = c.arrival_timestamp
                    self.in_connection[c.arrival_station] = i
                    self.trip_flag[c.trip_id] = 1


        if self.in_connection[arrival_station] == MAX_INT:
            return -1
        else:
            route = []

            last_connection_index = self.in_connection[arrival_station]

            while last_connection_index != MAX_INT:
                connection = conns[last_connection_index]
                route.append(connection)
                last_connection_index = self.in_connection[connection.departure_station]

            return route[::-1]


    def cmpr(self, routes):
        r_lst = []
        for r in routes:
            route2 = []
            temp = r[0]
            for c in r[1:]:
                if c.trip_id != temp.trip_id:
                    route2.append(c.departure_station)
                temp = c

            if len(route2) == 0:
                time_delta = r[-1].arrival_timestamp - r[0].departure_timestamp
                r_lst.append((time_delta, 0, -1, r[0].route_id, r[-1].route_id,
```

```
                r[0].departure_station, r[-1].arrival_station))
            elif r[-1].ori_station == r[0].departure_station:
                time_delta = r[-1].arrival_timestamp - r[-1].ori_dt
                r_lst.append((time_delta, 0, -1, r[0].route_id, r[-1].route_id,
                r[-1].ori_station, r[-1].arrival_station))
            else:
                time_delta = r[-1].arrival_timestamp - r[0].departure_timestamp
                temp = r[-1].ori_station
                if temp in route2:
                    if route2[-1]!=temp:
                        for k in range(len(route2)):
                            if route2[k] == temp:
                                dps = k
                                break

                        route2 = route2[0:dps]
                        route2.append(r[-1].ori_station)

                r_lst.append((time_delta, len(route2), route2[0], r[0].route_id,
                r[-1].route_id, r[0].departure_station, r[-1].arrival_station))


        #print r_lst
        pareto = OrderedSet()
        pareto.add((r_lst[0][0], r_lst[0][1]))


        for each in r_lst:
            dom_list = []
            for every in pareto:
                temp = each[0] - every[0]
                temp2 = each[1] - every[1]
                ind = 0
                if temp > 31 and temp2 >= 0:
                    ind = 1
                    break
                if temp < -31 and temp2 <= 0:
                    dom_list.append(every)
                if len(r_lst) > 2:
                    if -11 < temp < 32 and temp2 > 0:
                        ind = 1
                        break
                    if -32 < temp < 11 and temp2 < 0:
                        dom_list.append(every)

            if ind == 0:
                if len(dom_list) > 0:
                    for k in dom_list:
                        pareto.discard(k)

                pareto.add((each[0], each[1]))


        dom_list = OrderedSet()
        for each in r_lst:
            for every in pareto:
                if each[0] == every[0] and each[1] == every[1]:
                    dom_list.add(each[1:])

        for each in dom_list:
            self.results.append(each)

        #print dom_list



    def compute(self, departure_station, arrival_station):
```

```
        scan_points = range(600, 690, 10)

        if departure_station in self.souths and arrival_station in self.norths:
            status0 = 1
        elif departure_station in self.norths and arrival_station in self.souths:
            status0 = 1
        elif departure_station in self.norths and arrival_station in self.norths:
            status0 = 2
        else:
            status0 = 0

        #start_time = time()

        if status0 == 0:
            route_list = []
            for dt in scan_points:
                if len(route_list) > 0:
                    if dt <= route_list[-1][0].departure_timestamp:
                        continue

                self.in_connection = array('I', [MAX_INT for _ in range(MAX_STATIONS)])
                self.earliest_arrival = array('H', [MAX_INT2 for _ in range(MAX_STATIONS)])
                self.trip_flag = {k:0 for k in self.trips}
                self.earliest_arrival[departure_station] = dt - self.ch_time[departure_station]

                b_p = self.bpl[dt-3]

                bb = self.main_loop(self.timetable0.connections[b_p:], arrival_station)
                if bb != -1:
                    route_list.append(bb)

        if status0 == 2:
            route_list = []
            for dt in scan_points:
                if len(route_list) > 0:
                    if dt <= route_list[-1][0].departure_timestamp:
                        continue

                self.in_connection = array('I', [MAX_INT for _ in range(MAX_STATIONS)])
                self.earliest_arrival = array('H', [MAX_INT2 for _ in range(MAX_STATIONS)])
                self.trip_flag = {k:0 for k in self.trips}
                self.earliest_arrival[departure_station] = dt - self.ch_time[departure_station]

                b_p = self.bps[dt-3]

                bb = self.main_loop(self.timetable1.connections[b_p:], arrival_station)
                if bb != -1:
                    route_list.append(bb)

        if status0 == 1:
            route_list = []
            for dt in scan_points:
                if len(route_list) > 0:
                    if dt <= route_list[-1][0].departure_timestamp:
                        continue

                self.in_connection = array('I', [MAX_INT for _ in range(MAX_STATIONS)])
                self.earliest_arrival = array('H', [MAX_INT2 for _ in range(MAX_STATIONS)])
                self.trip_flag = {k:0 for k in self.trips}
                self.earliest_arrival[departure_station] = dt - self.ch_time[departure_station]
```

```

                b_p = self.bpl[dt-3]

                bb = self.main_loop(self.timetable0.connections[b_p:], arrival_station)
                if bb != -1:
                    route_list.append(bb)

            route_list2 = []
            for dt in scan_points:
                if len(route_list2) > 0:
                    if dt <= route_list2[-1][0].departure_timestamp:
                        continue

                self.in_connection = array('I', [MAX_INT for _ in range(MAX_STATIONS)])
                self.earliest_arrival = array('H', [MAX_INT2 for _ in range(MAX_STATIONS)])
                self.trip_flag = {k:0 for k in self.trips}
                self.earliest_arrival[departure_station] = dt - self.ch_time[departure_station]

                b_p = self.bps[dt-3]

                bb = self.main_loop(self.timetable1.connections[b_p:], arrival_station)
                if bb != -1:
                    route_list2.append(bb)

            route_list.extend(route_list2)


        #end_time = time()
        #start_time = round(end_time - start_time,7)

        if len(route_list) > 0:
            self.cmpr(route_list)
            #self.counters += 1

        #print str(start_time) + 's \n'

    #def get_counts(self):
        #print self.counters
    def combi(self):
        return self.results


def main():
    csa = CSA()

    temp = [line.strip().split(',') for line in open('cr-30.csv') if not line.startswith('trans')]
    od_pair = [(int(each[6]), int(each[7])) for each in temp]
    start_time = time()
    for y in od_pair:
        csa.compute(y[0], y[1])

    end_time = time()
    start_time = round(end_time - start_time,7)

    results = csa.combi()
    results = pd.DataFrame(results, columns=['ch_num', 'ts_2', 'ru_f', 'ru_c', 'ori_2', 'des_2'])
    results.to_csv('shortest-paths.csv', index=False)
    print str(start_time) + 's '


    '''
    sss = [sample(xrange(2583),2) for j in range(1000)]
```

```

    start_time = time()
    for each in sss:
        csa.compute(each[0], each[1])

    csa.get_counts()
    end_time = time()
    start time = (end time - start time)/1000
    start_time = round(start_time, 7)
    print start_time  '''


    '''
    while True:
        tt = raw_input('Origin and Destination <CRS,CRS> : ').split(',')
        if len(tt) == 1 and tt[0].strip() == '':
            print '\n No parameter provided. Program has stopped. \n'
            break
        elif len(tt) < 2:
            print '\n Less than 3 parameters provided. Enter again... \n'
            continue
        else:
            tt[0] = csa.dict3[tt[0].upper()]
            tt[1] = csa.dict3[tt[1].upper()]


            csa.compute(tt[0], tt[1]) '''




if __name__ == '__main__':
    main()



data = pd.read_csv('critical-routes-3.csv', usecols=['trans', 'ru_f', 'ru_c', 'ori_f',
'des_c'])

data2 = pd.read_csv('stop-dicts.txt')

  = data2.rename(columns={'parent station':'trans', 's num':'ts 2'}, inplace=True)

data3 = data2[['trans', 'ts_2']]

data = pd.merge(data, data3, how='inner', on='trans')

  = data2.rename(columns={'s name':'ori f', 'ts 2':'ori 2'}, inplace=True)

data3 = data2[['ori_f', 'ori_2']]

data = pd.merge(data, data3, how='inner', on='ori_f')

_ = data2.rename(columns={'ori_f':'des_c', 'ori_2':'des_2'}, inplace=True)

data3 = data2[['des_c', 'des_2']]

data = pd.merge(data, data3, how='inner', on='des_c')

data.to_csv('cr-3.csv', index=False)



data = pd.read_csv('cr-3.csv')

data = data.assign(ods=lambda x: x['ori_2']*10000 + x['des_2'])

```

```
temp1 = data['ods'].value_counts()
temp2 = list(temp1.index)
temp3 = list(temp1.values)
temp1 = pd.DataFrame({'ods':temp2, 'counts':temp3})
data = pd.merge(data, temp1, on='ods')
data.to_csv('intermediate-20.csv', index=False)
data = data[data['counts']==1]
data.to_csv('cr-30.csv', index=False)


data = pd.read_csv('shortest-paths.csv')
data = data.drop_duplicates()
data = data.assign(ods=lambda x: x['ori_2']*10000 + x['des_2'])
temp1 = data['ods'].value_counts()
temp2 = list(temp1.index)
temp3 = list(temp1.values)
temp1 = pd.DataFrame({'ods':temp2, 'counts':temp3})
data = pd.merge(data, temp1, on='ods')
#data.to_csv('intermediate-22.csv', index=False)
data = data[data['ch_num']==1]
#data = data.drop_duplicates(['ods'])
del data['ch_num']
data.to_csv('shortest-paths-2.csv', index=False)


data1 = pd.read_csv('shortest-paths-2.csv')
data =  pd.read_csv('cr-30.csv', usecols=['trans', 'ru_f', 'ru_c', 'ori_f', 'des_c',
'ts_2', 'ori_2', 'des_2'])
data = data.reindex(columns=['trans', 'ori_f', 'des_c', 'ts_2', 'ru_f', 'ru_c',
'ori_2', 'des_2'])
data = pd.merge(data, data1, on=['ts_2', 'ru_f', 'ru_c', 'ori_2', 'des_2'])
data.to_csv('cr-4.csv', index=False)


data = pd.read_csv('critical-routes-3.csv', usecols=['trans', 'ru_f', 'ru_c', 'serv_f'])
data1 = pd.read_csv('cr-5.csv', usecols=['trans', 'ori_f', 'des_c', 'ru_f', 'ru_c'])
data = pd.merge(data, data1, how='inner', on=['trans', 'ru_f', 'ru_c'])
data = data.reindex(columns=['trans', 'ru_f', 'ru_c', 'serv_f', 'ori_f', 'des_c'])
```

```
data.to_csv('intermediate-5.csv', index=False)


L1 = len(data)
L2 = []

for i in range(L1-1):
    for j in range(i+1,L1):
        if data.iloc[i,0]==data.iloc[j,0]:
            continue
        #if data.iloc[i,3]!=data.iloc[j,3]:
            #continue
        t1 = (data.iloc[i,2]==data.iloc[j,1]) & (data.iloc[i,1]!=data.iloc[j,2])
        t2 = (data.iloc[i,2]!=data.iloc[j,1]) & (data.iloc[i,1]==data.iloc[j,2])
        if t1:
            temp = [data.iloc[i,1], data.iloc[i,2], data.iloc[i,3], data.iloc[i,0],
            data.iloc[i,4], data.iloc[i,5], data.iloc[j,1], data.iloc[j,2],
            data.iloc[j,3], data.iloc[j,0], data.iloc[j,5]]
            L2.append(temp)
        if t2:
            temp = [data.iloc[j,1], data.iloc[j,2], data.iloc[j,3], data.iloc[j,0],
            data.iloc[j,4], data.iloc[j,5], data.iloc[i,1], data.iloc[i,2],
            data.iloc[i,3], data.iloc[i,0], data.iloc[i,5]]
            L2.append(temp)

if len(L2)==0:
    print 'No Critical Routes with more than one transfer found! Empty candidate list!'
else:
    L2 = pd.DataFrame(L2, columns=['r_f1', 'r_c1', 'serv_1', 't_1', 'o_1', 'd_1',
    'r_f2', 'r_c2', 'serv_2', 't_2', 'd_2'])
    L2.to_csv('crs-1plus.csv', index=False)



data = pd.read_csv('intermediate-5.csv')

data2 = pd.read_csv('calendar-2.csv', usecols=['service_id', 's_date', 'e_date',
'diff2'])

_ = data2.rename(columns={'service_id': 'serv_f'}, inplace=True)

data = pd.merge(data, data2, on='serv_f')

data.to_csv('cr-6.csv', index=False)
```

Appendix C

Real-world examples of the potential effect of applying the method of increasing MTT to critical routes

[Note: MTT = Minimum Transfer Time. ECB = Ebbw Vale Town – Cardiff Central – Birmingham New Street. KWN = Knottingley – Wakefield Kirkgate – Nottingham.]

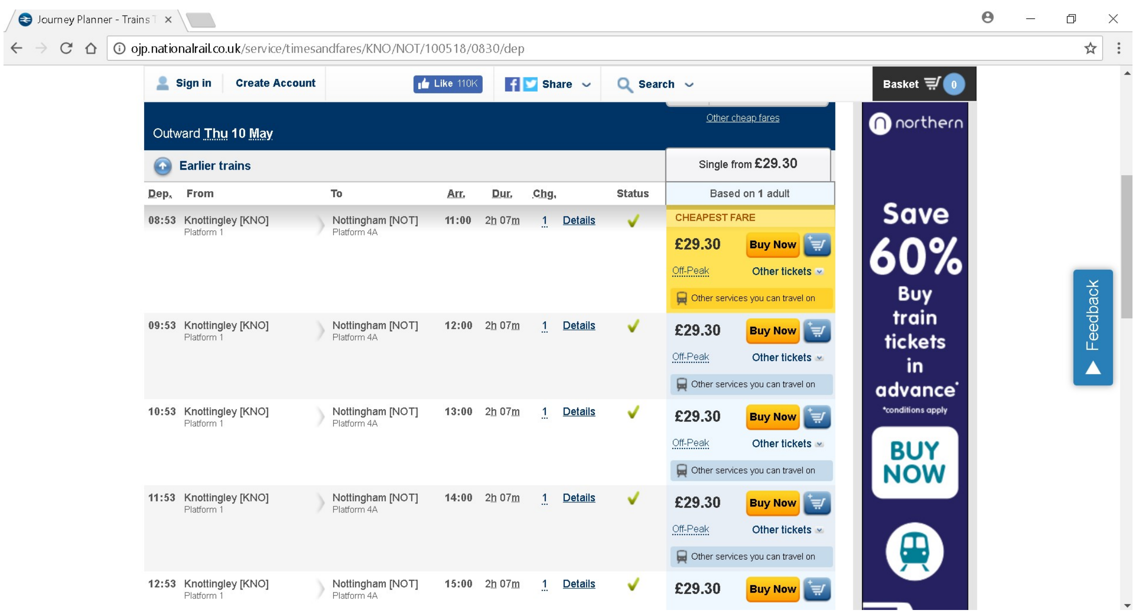


Figure C1 KWN – recommendations based on the default MTT (4 mins)

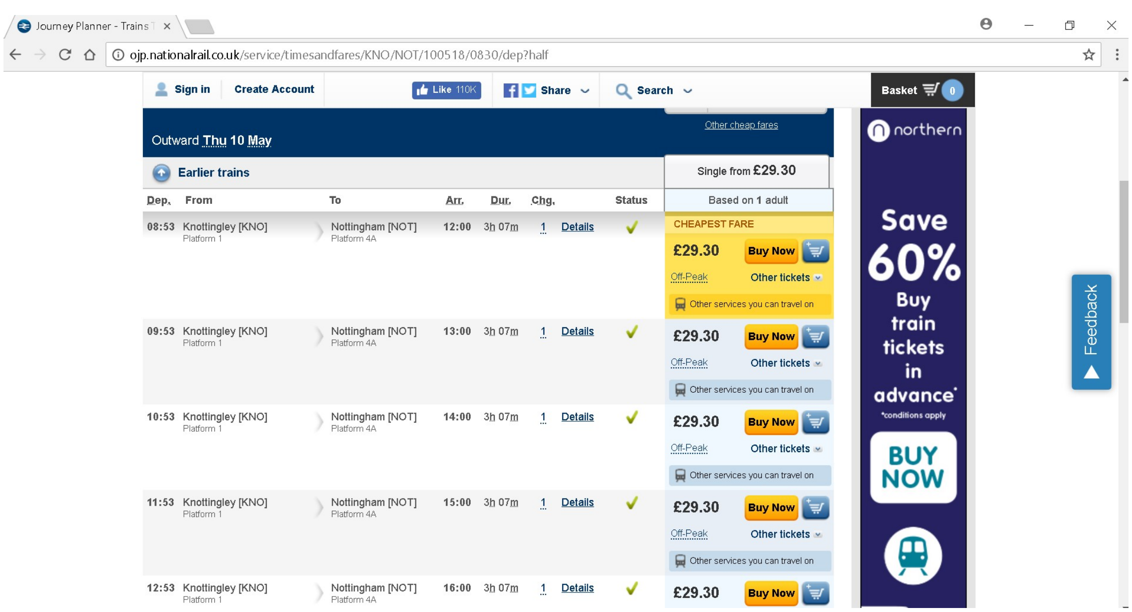


Figure C2 KWN – recommendations based on an increased MTT (>7 and <=67 mins)

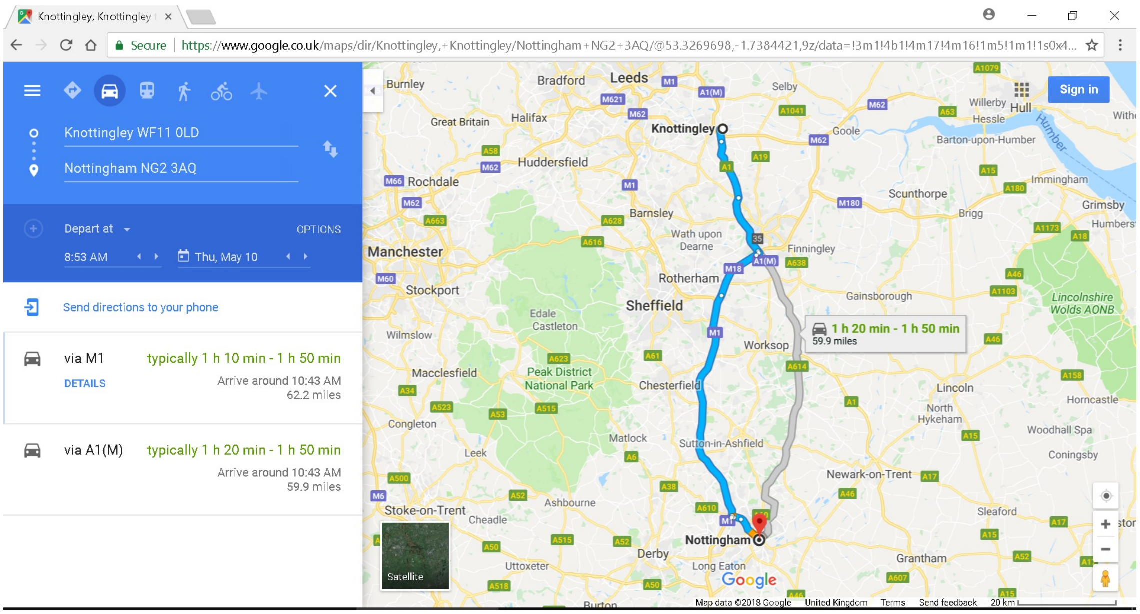


Figure C3 KWN – estimated journey time by car

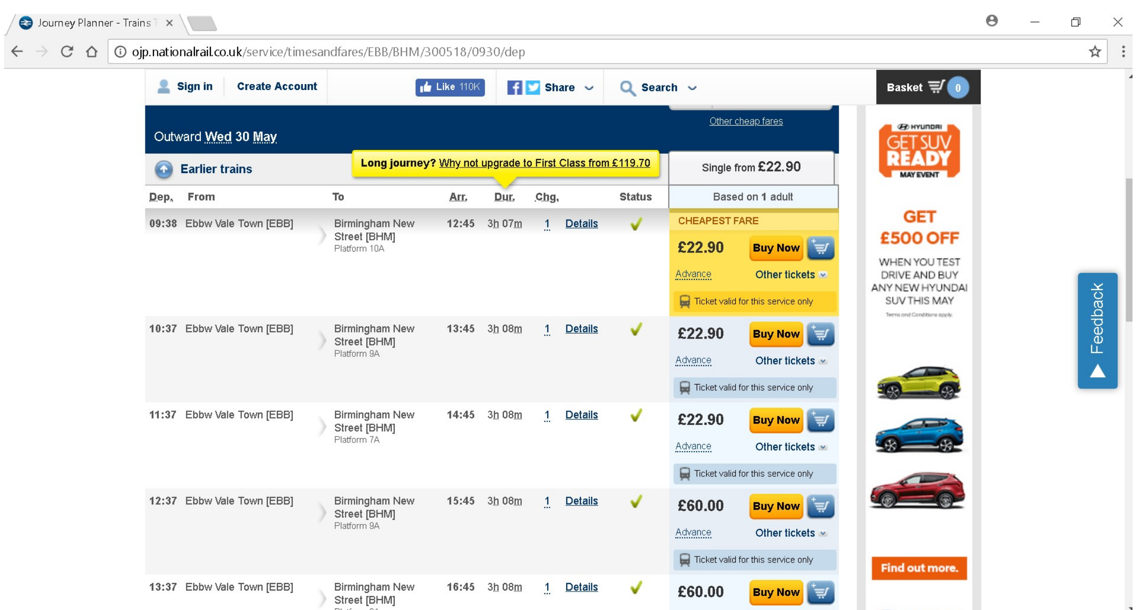


Figure C4 ECB – recommendations based on the default MTT (7 mins)

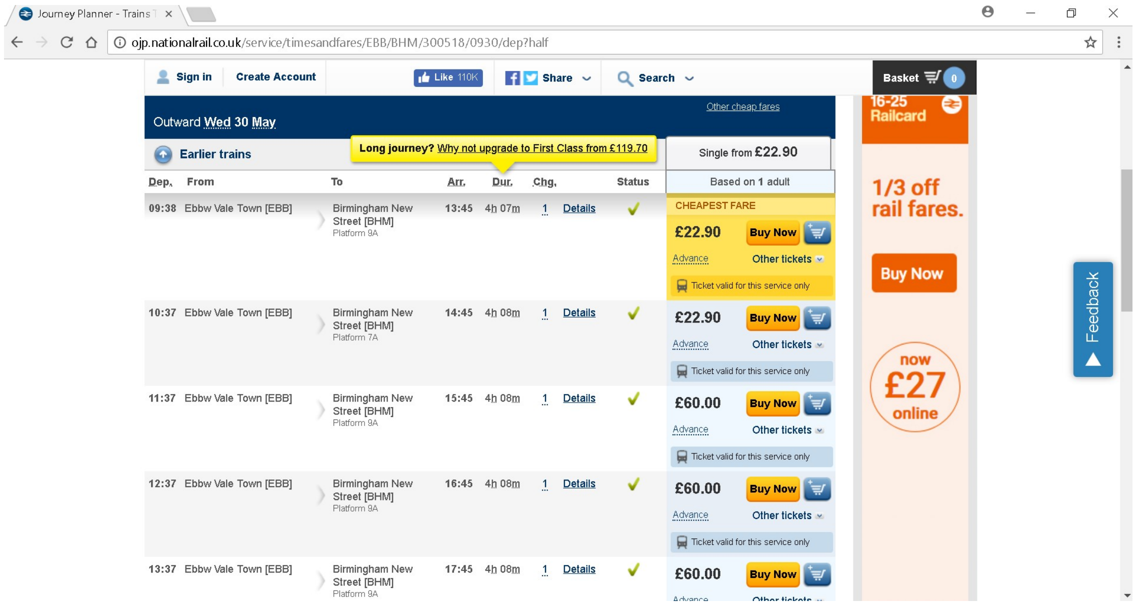


Figure C5 ECB – recommendations based on an increased MTT (>8 and <=68 mins)

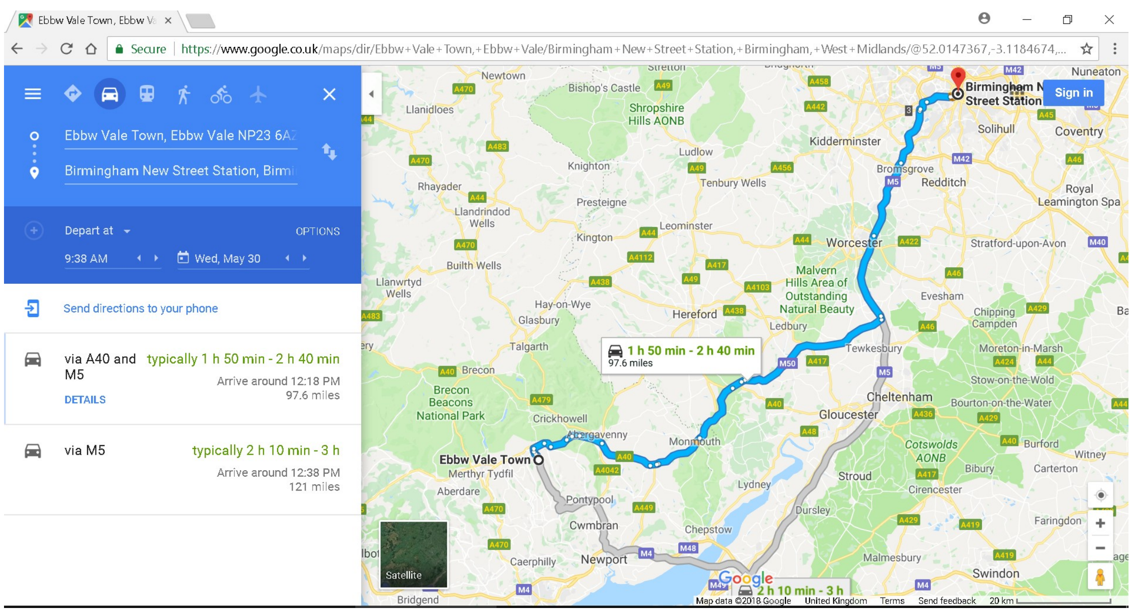


Figure C6 ECB – estimated journey time by car

Appendix D

A Python implementation of the back end of IPS and PBPM

Note: the source codes are grouped (by general functionality) into three files named ips-pbpm-1.py, ips-23.py, and pbpm-234.py, respectively. Generally, the file named ips-pbpm-1.py can be thought of as the implementation of Step 1 in IPS and PBPM (c.f. Subsection 4.3.2), ips-23.py the implementation of Steps 2 – 3 in IPS, and pbpm-234.py the implementation of Steps 2 – 4 in PBPM. Three parameters – sample size, net transfer time, and threshold for arrival lateness – have been set as variables to facilitate the comparison between different choices of parameters. The example route in this implementation is Liverpool Lime Street – Manchester Piccadilly – Doncaster. Only minor modifications needed to apply the codes to other critical routes.

(1) ips-pbpm-1.py

```python
from pandas import DataFrame, Series
import pandas as pd
import json
from datetime import datetime
import numpy as np
import os


## author: Yiwei Guo  30/07/2016 @soton


L2 = []
cdir = os.getcwd()
ttt=os.walk(cdir)

for root, dirs, files in ttt:
    for d in dirs:
        path = os.path.join(root, d)
        L2.append(path)

for each in L2:
    os.chdir(each)
    cdir = os.getcwd()
    ttt = os.listdir(cdir)

    for file in ttt:
        data = json.load(open(file))
        data = data["services"]
        L = len(data)

        keys = ["serviceUid", "runDate", "atocCode", "serviceType"]
        info = DataFrame(data,columns=keys)
        rundate = info.iat[L/2, 1]

        list1 = []

        for i in range(L):
            data1 = data[i]["locationDetail"]
            list1.append(data1)

        keys = ["tiploc", "gbttBookedArrival", "gbttBookedDeparture",
        "realtimeArrival", "realtimeDeparture", "displayAs", "platform"]
        data1 = DataFrame(list1,columns=keys)

        tiploc = data1.iat[0,0]
        tiploc = str(tiploc)
        rundate = str(rundate)
        rundate = datetime.strptime(rundate, '%Y-%m-%d')
        rundate = rundate.strftime('-%a-%d-%m-%Y')
        if 'arr' in file:
            file = 'ARR-' + tiploc + rundate + '.csv'
        else:
            file = tiploc + rundate + '.csv'

        info = pd.concat([info, data1], axis=1)
        info.to_csv(file,index=False)

    cdir = os.getcwd()
    ttt = os.listdir(cdir)

    for file in ttt:
        if file.endswith('.csv'):
            if file.startswith('MNCRPIC'):
                data = pd.read_csv(file)
                name = file
            elif file.startswith('LVRPL'):
                info1 = pd.read_csv(file)
```

```
            else:
                info2 = pd.read_csv(file)


    L1 = ['Y00256', 'Y00255', 'Y00254', 'Y00253', 'Y00252']
    L2 = ['Y00300', 'Y00298', 'Y00296', 'Y00294', 'Y00292']

    flag = info1['serviceUid'].isin(L1)
    info1 = info1[flag]
    info1 = info1[['serviceUid', 'tiploc', 'displayAs']]

    flag = info2['serviceUid'].isin(L2)
    info2 = info2[flag]

    data = data[(data['gbttBookedDeparture']>=630) & (data['gbttBookedDeparture']<=2230)]

    serviceUid = list(data['serviceUid'].values)
    runDate = list(data['runDate'].values)
    atocode = list(data['atocCode'].values)
    tiploc = list(data['tiploc'].values)
    display = list(data["displayAs"].values)
    plt = list(data["platform"].values)
    rtarr = data["realtimeArrival"].values
    rtarr = rtarr.astype(np.string_)
    pubarr = data["gbttBookedArrival"].values
    pubarr = pubarr.astype(np.string_)
    rtdep = data["realtimeDeparture"].values
    rtdep = rtdep.astype(np.string_)
    pubdep = data["gbttBookedDeparture"].values
    pubdep = pubdep.astype(np.string_)

    rtarrival = []
    for every in rtarr:
        if every=='[' or every==']':
            continue
        elif every=='nan':
            rtarrival.append(None)
        elif every.endswith('.0'):
            every = datetime.strptime(every,'%H%M.0')
            every = every.strftime('%H:%M')
            rtarrival.append(every)
        else:
            every = datetime.strptime(every,'%H%M')
            every = every.strftime('%H:%M')
            rtarrival.append(every)

    rtarr_cov = []
    for each in rtarrival:
        if each is None:
            rtarr_cov.append(np.nan)
        else:
            hour = int(each[:2])
            minute = int(each[3:])
            converted = hour*60 + minute
            rtarr_cov.append(converted)

    rtdepature = []
    for every in rtdep:
        if every=='[' or every==']':
            continue
        elif every=='nan':
            rtdepature.append(None)
        elif every.endswith('.0'):
            every = datetime.strptime(every,'%H%M.0')
            every = every.strftime('%H:%M')
            rtdepature.append(every)
        else:
```

```
            every = datetime.strptime(every,'%H%M')
            every = every.strftime('%H:%M')
            rtdepature.append(every)

    rtdep_cov = []
    for each in rtdepature:
        if each is None:
            rtdep_cov.append(np.nan)
        else:
            hour = int(each[:2])
            minute = int(each[3:])
            converted = hour*60 + minute
            rtdep_cov.append(converted)

    pubarrival = []
    for every in pubarr:
        if every=='[' or every==']':
            continue
        elif every=='nan':
            pubarrival.append(None)
        elif every.endswith('.0'):
            every = datetime.strptime(every,'%H%M.0')
            every = every.strftime('%H:%M')
            pubarrival.append(every)
        else:
            every = datetime.strptime(every,'%H%M')
            every = every.strftime('%H:%M')
            pubarrival.append(every)

    pubarr_cov = []
    for each in pubarrival:
        if each is None:
            pubarr_cov.append(np.nan)
        else:
            hour = int(each[:2])
            minute = int(each[3:])
            converted = hour*60 + minute
            pubarr_cov.append(converted)

    pubdepature = []
    for every in pubdep:
        if every=='[' or every==']':
            continue
        elif every=='nan':
            pubdepature.append(None)
        elif every.endswith('.0'):
            every = datetime.strptime(every,'%H%M.0')
            every = every.strftime('%H:%M')
            pubdepature.append(every)
        else:
            every = datetime.strptime(every,'%H%M')
            every = every.strftime('%H:%M')
            pubdepature.append(every)

    pubdep_cov = []
    for each in pubdepature:
        if each is None:
            pubdep_cov.append(np.nan)
        else:
            hour = int(each[:2])
            minute = int(each[3:])
            converted = hour*60 + minute
            pubdep_cov.append(converted)

    arr_lateness = []
    count = 0
    while count < len(rtarr_cov):
```

```
        flag = rtarr_cov[count] - pubarr_cov[count]
        arr_lateness.append(flag)
        count = count + 1


    dep_lateness = []
    count = 0
    while count < len(rtdep_cov):
        flag = rtdep_cov[count] - pubdep_cov[count]
        dep_lateness.append(flag)
        count = count + 1

    data = DataFrame({'serviceUid':serviceUid,'runDate':runDate,'atoc':atocode,'tiploc':tiploc,'displayAs':display,'platform':plt,'pubarr_cov':pubarr_cov,'pubdep_cov':pubdep_cov,'rtarr_cov':rtarr_cov,'rtdep_cov':rtdep_cov,'arr_lateness':arr_lateness,'dep_lateness':dep_lateness},columns=['serviceUid','runDate','atoc','tiploc','displayAs','platform','pubarr_cov','pubdep_cov','rtarr_cov','rtdep_cov','arr_lateness','dep_lateness'])
    name = name.replace('MNCRPIC', 'Route1')
    data1 = data['serviceUid'].isin(L1)
    data1 = data[data1]
    data1 = pd.merge(data1, info1, how='outer',on='serviceUid')
    data1.to_csv(name,index=False)


    serviceUid = list(info2['serviceUid'].values)
    tiploc = list(info2['tiploc'].values)
    display = list(info2["displayAs"].values)
    rtarr = info2["realtimeArrival"].values
    rtarr = rtarr.astype(np.string_)
    pubarr = info2["gbttBookedArrival"].values
    pubarr = pubarr.astype(np.string_)

    rtarrival = []
    for every in rtarr:
        if every=='[' or every==']':
            continue
        elif every=='nan':
            rtarrival.append(None)
        elif every.endswith('.0'):
            every = datetime.strptime(every,'%H%M.0')
            every = every.strftime('%H:%M')
            rtarrival.append(every)
        else:
            every = datetime.strptime(every,'%H%M')
            every = every.strftime('%H:%M')
            rtarrival.append(every)

    rtarr_cov = []
    for each in rtarrival:
        if each is None:
            rtarr_cov.append(np.nan)
        else:
            hour = int(each[:2])
            minute = int(each[3:])
            converted = hour*60 + minute
            rtarr_cov.append(converted)

    pubarrival = []
    for every in pubarr:
        if every=='[' or every==']':
            continue
        elif every=='nan':
            pubarrival.append(None)
        elif every.endswith('.0'):
            every = datetime.strptime(every,'%H%M.0')
            every = every.strftime('%H:%M')
```

```
            pubarrival.append(every)
        else:
            every = datetime.strptime(every,'%H%M')
            every = every.strftime('%H:%M')
            pubarrival.append(every)

    pubarr_cov = []
    for each in pubarrival:
        if each is None:
            pubarr_cov.append(np.nan)
        else:
            hour = int(each[:2])
            minute = int(each[3:])
            converted = hour*60 + minute
            pubarr_cov.append(converted)

    arr lateness = []
    count = 0
    while count < len(rtarr_cov):
        flag = rtarr_cov[count] - pubarr_cov[count]
        arr_lateness.append(flag)
        count = count + 1

    info2 =
    DataFrame({'serviceUid':serviceUid,'tiploc':tiploc,'displayAs':display,'arr lateness'
    :arr_lateness},columns=['serviceUid','tiploc','displayAs','arr_lateness'])
    name = name.replace('Route1', 'Route2')
    data1 = data['serviceUid'].isin(L2)
    data1 = data[data1]
    data1 = pd.merge(data1, info2, how='outer',on='serviceUid')
    data1.to_csv(name,index=False)
```

(2) ips-23.py

```python
from pandas import Series,DataFrame
import pandas as pd
import os


## author: Yiwei Guo  30/07/2016


#group + trim + count-canc
cdir = os.getcwd()
fnames = os.listdir(cdir)
route1 = []
route2 = []

for file in fnames:
    if file.startswith('Route1'):
        route1.append(file)
    if file.startswith('Route2'):
        route2.append(file)

frames1 = []
frames2 = []

count1 = {'Y00256':0,'Y00255':0,'Y00254':0,'Y00253':0,'Y00252':0}
count2 = {'Y00300':0,'Y00298':0,'Y00296':0,'Y00294':0,'Y00292':0}

count1 = Series(count1)
count2 = Series(count2)

for each in route1:
    data = pd.read_csv(each)
    List1 = count1.index
    List1 = Series(List1)
    if len(data['serviceUid']) < len(count1):
        List2 = data['serviceUid'].values
        List2 = list(List2)
        mask = List1.isin(List2)
        mask2 = mask[mask==False]
        mask = mask2.index
        List2 = List1[mask]
        for every in List2:
            count1[every] += 1

    mask2 = data[(data['displayAs_x'] == 'CANCELLED_CALL') | (data['displayAs_y'] ==
    'CANCELLED_CALL')]
    mask = mask2.index
    List1 = List1[mask]
    for every in List1:
        count1[every] += 1

    mask = data[(data['displayAs_x'] != 'CANCELLED_CALL') & (data['displayAs_y'] !=
    'CANCELLED_CALL')]
    frames1.append(mask)

data = pd.concat(frames1)
data.to_csv('result1.csv',index=False)
count1.to_csv('canc-1.csv', index=True)

for each in route2:
    data = pd.read_csv(each)
    List1 = count2.index
    List1 = Series(List1)
    if len(data['serviceUid']) < len(count2):
        List2 = data['serviceUid'].values
        List2 = list(List2)
        mask = List1.isin(List2)
        mask2 = mask[mask==False]
```

```
        mask = mask2.index
        List2 = List1[mask]
        for every in List2:
            count2[every] += 1

    mask2 = data[(data['displayAs_x'] == 'CANCELLED_CALL') | (data['displayAs_y'] == 'CANCELLED_CALL')]
    mask = mask2.index
    List1 = List1[mask]
    for every in List1:
        count2[every] += 1

    mask = data[(data['displayAs_x'] != 'CANCELLED_CALL') & (data['displayAs_y'] != 'CANCELLED_CALL')]
    frames2.append(mask)

data = pd.concat(frames2)
data.to_csv('result2.csv',index=False)
count2.to_csv('canc-2.csv', index=True)

#merge
frames1 = pd.read_csv('result1.csv', usecols = ['serviceUid', 'runDate','tiploc_x','platform','pubarr_cov','rtarr_cov'])
frames2 = pd.read_csv('result2.csv', usecols = ['serviceUid', 'runDate','tiploc_x','platform','pubdep_cov','rtdep_cov','tiploc_y','arr_lateness_y'])

data = pd.merge(frames1,frames2,how='inner',on='tiploc_x')

data.to_csv('merged.csv',index=False)

#filter uncorr + filter corr
data = data.assign(window = lambda x: x['pubdep_cov'] - x['pubarr_cov'])
data = data[data.window < 13]
data = data[data.window > 7]

data.to_csv('filter_uncorr.csv',index=False)

data = data[data['runDate_x']==data['runDate_y']]
data.to_csv('filter_corr.csv',index=False)


#stats:TS=m
n = raw_input('The number of weeks to calculate: ')
n = int(n)
m = raw_input('Net transfer time is set to: ')
m = int(m)
k = raw_input('Maximum lateness at destination: ')
k = int(k)

data = pd.read_csv('filter_corr.csv')
data1 = data[['serviceUid_x','serviceUid_y']]
data1 = data1.drop_duplicates()

route1 = list(data1['serviceUid_x'].values)
route2 = list(data1['serviceUid_y'].values)
rb = []

for trip in route1:
    data2 = data[data['serviceUid_x']==trip]
    a = count1[trip]
    b = data2['serviceUid_y'].drop_duplicates()
    b = list(b.values)
    b = count2[b[0]]
    a = max(a,b)
    count_all = len(data2) + a
    data2 = data2.assign(window2 = lambda x: x['rtdep_cov'] - x['rtarr_cov'])
    data2 = data2[(data2.window2 >= m) & (data2.arr_lateness_y < k)]
```

```
    count_suc = len(data2)
    rel = float(count_suc)/count_all
    rel = round(rel*100)
    rb.append(rel)

data1 = DataFrame({'route1':route1,'route2':route2,'rel-corr':rb},columns=['route1','route2','rel-corr'])
data1.to_csv('rel-correlated.csv',index=False)

data = pd.read_csv('filter_uncorr.csv')
data1 = data[['serviceUid_x','serviceUid_y']]
data1 = data1.drop_duplicates()

route1 = list(data1['serviceUid_x'].values)
route2 = list(data1['serviceUid_y'].values)
rb = []

for trip in route1:
    data2 = data[data['serviceUid_x']==trip]
    a = count1[trip]
    b = data2['serviceUid_y'].drop_duplicates()
    b = list(b.values)
    b = count2[b[0]]
    count_all = len(data2) + (a + b)*n*5 - (a*b)
    data2 = data2.assign(window2 = lambda x: x['rtdep_cov'] - x['rtarr_cov'])
    data2 = data2[(data2.window2 >= m) & (data2.arr_lateness_y < k)]
    count_suc = len(data2)
    rel = float(count_suc)/count_all
    rel = round(rel*100)
    rb.append(rel)

data1 = DataFrame({'route1':route1,'route2':route2,'rel-uncorr':rb},columns=['route1','route2','rel-uncorr'])
data1.to_csv('rel-uncorr.csv',index=False)


data1 = pd.read_csv('rel-correlated.csv')
data2 = pd.read_csv('rel-uncorr.csv')

result = pd.merge(data1,data2)

result = result.assign(rel_mean = lambda x: (x['rel-corr']+x['rel-uncorr'])/2, rel_linear = lambda x: 0.2*x['rel-corr']+0.8*x['rel-uncorr'])
result.to_csv('rel-table-man.csv',index=False)
```

(3) pbpm-234.py

```python
from pandas import Series,DataFrame
import pandas as pd
import os


## author: Yiwei Guo  30/07/2016 @soton


#group + trim + count-canc

n = raw_input('The number of weeks to calculate: ')
n = int(n)*5

cdir = os.getcwd()
fnames = os.listdir(cdir)
route1 = []
route2 = []

for file in fnames:
    if file.startswith('Route1'):
        route1.append(file)
    if file.startswith('Route2'):
        route2.append(file)

frames1 = []
frames2 = []

count1 = {'Y70205':n,'Y70208':n,'Y80803':n,'Y70177':n,'Y70216':n,'Y70219':n}

#count2 =
{'W83537':0,'W83538':0,'W83539':0,'W83540':0,'W83541':0,'W83542':0,'W83543':0,'W83544':0,
'W83545':0,'P01078':0}

count1 = Series(count1)
#count2 = Series(count2)

for each in route1:
    data = pd.read_csv(each)
    frames1.append(data)

data = pd.concat(frames1)


L1 = data['serviceUid'].drop_duplicates()
L1 = list(L1.values)

for every in L1:
    data2 = data[data['serviceUid']==every]
    temp = len(data2)
    if temp < n:
        data1 = data2[data2['display_1']=='CANCELLED_CALL']
        if data1.empty == False:
            temp -= len(data1)
        count1[every] = temp


data = data[(data['displayAs'] != 'CANCELLED_CALL') & (data['display_1'] !=
'CANCELLED_CALL')]
data.to_csv('result1.csv',index=False)
count1.to_csv('count-1.csv', index=True)

for each in route2:
    data = pd.read_csv(each)
    frames2.append(data)

data = pd.concat(frames2)
data = data[(data['displayAs'] != 'CANCELLED_CALL') & (data['display_3'] !=
```

```
    'CANCELLED_CALL')]

data.to_csv('result2.csv',index=False)




#merge
frames1 = pd.read_csv('result1.csv')
frames2 = pd.read_csv('result2.csv')

data = pd.merge(frames1,frames2,on=['tiploc'])

data.to_csv('merged.csv',index=False)



#filter_uncorr + filter_corr
data = data.assign(window = lambda x: x['gbttBookedDeparture'] - x['gbttBookedArrival'])
data = data[data.window < 12]
data = data[data.window > 8]

data = data[data['runDate_x']==data['runDate_y']]
data = data.sort_values(by=['runDate_x', 'gbttBookedArrival'])



m = raw_input('Net transfer time is set to: ')
m = int(m)
# k = raw_input('Maximum lateness at destination: ')
# k = int(k)

#data = pd.read_csv('filter_corr.csv')

data = data.assign(window2 = lambda x: x['realtimeDeparture'] - x['realtimeArrival'])
data = data.assign(arrlate_3 = lambda x: x['rtarr_3'] - x['pubarr_3'])

data.to_csv('filter_corr.csv',index=False)

data1 = data[['serviceUid_x','serviceUid_y']]
data1 = data1.drop_duplicates()

route1 = list(data1['serviceUid_x'].values)
route2 = list(data1['serviceUid_y'].values)
rb = []
frames1 = []
frames2 = []

for trip in route1:
    data2 = data[data['serviceUid_x']==trip]
    a = count1[trip]

    count_all = a
    #data2 = data2.assign(window2 = lambda x: x['rtdep_cov'] - x['rtarr_cov'])
    data2 = data2[data2.window2 >= m]
    count_suc = len(data2)
    rel = float(count_suc)/count_all
    rel = round(rel*100)
    frames1.append(count_suc)
    frames2.append(count_all)
    rb.append(rel)

data1 = DataFrame({'route1':route1,'route2':route2,'rel-corr':rb, 'count-suc':frames1,
'count-all':frames2},columns=['route1','route2','rel-corr','count-suc','count-all'])
data1.to_csv('rel-correlated.csv',index=False)
```

```
frames1 = {}

for each in route2:
    data2 = data[data['serviceUid_y']==each]
    rb = data2['arrlate_3'].mean()
    frames1[each] = rb

frames1 = Series(frames1)
frames1.to_csv('mean-late.csv', index=True)
```