Integer Programming Models and Algorithms for Rail Transport Rescheduling

March 2016

Keisuke Sato

Integer Programming Models and Algorithms for Rail Transport Rescheduling

Graduate School of Systems and Information Engineering
University of Tsukuba

March 2016

Keisuke Sato

Abstract

Rail transport systems in the greater part of Japan and the world are considerably large and highly complex at the present day. They are, therefore, operated in accordance with a set of predetermined schedules. Rail transport scheduling, which consists mainly of train timetabling, rolling stock assignment to train tasks, crew assignment to train tasks and rolling stock unit shunting, has been one of major applications of operations research and mathematical optimization, due to its complexity and significant impact on the operating costs and the service level to passengers and freight customers.

Railway operations, planned in any manner, are subject to internal or external disturbance. When they are disrupted to a certain extent, a rail transport management staff called dispatchers inevitably reschedules the plans to manage the situation. Rail transport rescheduling is a real-time process, and that as well as the problem scale and complexity derived from the scheduling phase makes the task more strenuous.

This thesis provides optimization approaches to the rail transport rescheduling to assist the dispatchers in their disruption management. We focus on passenger-oriented rescheduling of a passenger train timetable on a railway line with an ordinary track layout in Japan, operator-oriented rolling stock reassignment to passenger or freight train tasks on a railway network and operator-oriented crew reassignment to passenger or freight train tasks on a railway network. We model the rescheduling as integer programming problems, and propose algorithms to provide desirable solutions in real time.

In our timetable rescheduling formulation, we simultaneously model train operations and passengers' behavior to quantify and minimize our passenger-oriented objective function named further inconvenience to passengers. We apply column generation to the rolling stock and the crew rescheduling to obtain good solutions for them in real time. In the rolling stock rescheduling, efficient column generation and relaxation of the problem named set-covering relaxation are also introduced to enhance the computation.

We observe that our timetable rescheduling approach improves a train-punctuality-oriented rescheduled timetable within allowable computation time for a line with medium train traffic and delay. The rolling stock and the crew rescheduling algorithms provide optimal or near-optimal solutions for a practical network in acceptable time.

Acknowledgements

This thesis summarizes the research that I tackled during the past ten years at Railway Technical Research Institute (RTRI). First and foremost, I would like to express my deepest appreciation to my principal advisor Prof. Yoshitsugu Yamamoto, University of Tsukuba, for his continuous supervision and encouragement, for a total of five years after he began to advise me when I was an undergraduate student, and even for years when I did not take any course at the university.

I acknowledge a lot of valuable advice from my co-advisors Prof. Maiko Shigeno and Prof. Masahiro Hachimori, University of Tsukuba. I would also like to express my gratitude to the other doctoral thesis examination committee members Prof. Norio Tomii, Chiba Institute of Technology, Prof. Yoshio Sano and Prof. Yusuke Kobayashi, University of Tsukuba, for their instructive comments.

The co-authors of the papers which constitute this thesis, Mr. Naoto Fukumura, RTRI, and Mr. Kei Tamura when he was a student in a Master's program at Chiba Institute of Technology, in addition to Prof. Tomii, helped me develop the core of the thesis. Research collaborators from Japan Freight Railway Company, The University of Tokyo and Advanced Traffic Systems Corporation gave me the opportunity to implement and evaluate my ideas. I am greatly indebted to them.

I would like to express my appreciation to RTRI for giving me the opportunity to complete the research, and to the members of the institute, particularly Dr. Chikara Hirai, Mr. Shigeto Hiraguri, Mr. Takashi Sakaguchi, Mr. Ryuji Tsuchiya and Dr. Koichi Goto during his career at RTRI, for their supervision and management. I also thank the past and present members of Transport Operation Systems, Signalling and Transport Information Technology Division, for their help.

Last but not least, I am sincerely grateful to my parents and my sister. Far from here, they have been supporting me through the years.

Keisuke Sato
Tokyo, January 2016

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Rail transport systems in the greater part of Japan and the world are considerably large and highly complex at the present day. They are, therefore, operated in accordance with a set of predetermined schedules. Rail transport scheduling consists mainly of train timetabling, rolling stock assignment to train tasks, crew assignment to train tasks and rolling stock unit shunting at rolling stock depots and certain stations. Each of them has been one of major applications of operations research and mathematical optimization, for both passenger trains and freight trains. One reason for this is that the complexity of the problems arouses requests for algorithmic aid. Another is that planned schedules have a significant impact on the operating cost as well as the service level to passengers and freight customers. Academic approaches in these fields have brought a great success to some railway operating organizations.

Railway operations, planned in any manner, are subject to internal or external disturbance. When they are disrupted to a certain extent, a rail transport management staff called dispatchers inevitably reschedules the plans to manage the situation. Rail transport rescheduling is a real-time process, and that as well as the problem scale and complexity derived from the scheduling phase makes the task more strenuous. On the basis of the successful results of the scheduling phase and progress on computing power as well as algorithms for operations research and mathematical optimization problems, the rescheduling process is receiving attention in the rail transport literature.

1.2 Motivation and Purpose

This thesis provides optimization approaches to the rail transport rescheduling to assist the dispatchers in their disruption management. We focus on more passenger-oriented rescheduling of a passenger train timetable on a railway line, compared to simple train-punctuality-oriented rescheduling, which has been more vigorously discussed in the railway timetable rescheduling literature. We have to evaluate influence of disruption to the passengers in detail and to make deterioration of a services level to them as little as possible. This is not merely one of missions of railway operating organizations as public services; it is also the problem of the passengers' faith in the organizations and will consequently influence their future use of the services. We also present rolling stock reassignment to passenger or freight train tasks on a railway network and crew reassignment to passenger or freight train tasks on a railway network. They should be operator-oriented; the scale of the actions to be rescheduled is preferred to be as small as possible, since the dispatchers in charge of the rolling stock and the crew rescheduling have to communicate the modified pieces from the current schedule in the new schedule to a relevant staff. They have to bear the heavy burden if the scale is large. We model the rescheduling as integer programming problems, and propose algorithms to provide desirable solutions in real time.

1.3 Contributions

This thesis makes original contributions to the rail transport rescheduling literature, by providing integer programming formulations and algorithms for timetable rescheduling, rolling stock rescheduling and crew rescheduling. We briefly summarizes them for each of the rescheduling topics.

We focus on the timetable rescheduling of passenger trains on a railway line based on (K.) Sato et al. (2013). We simultaneously handle change of passengers' behavior called passenger rerouting and five detailed rescheduling measures to manage disrupted train operations: reordering of trains, local rerouting (change of track assignment) of trains inside a station, retiming of trains, change of rolling stock assignment to a train and change of a train type. This approach is significant for the passenger-oriented rescheduling since train orders affect passengers' behavior. With all these being incorporated into one mixed integer programming problem, we try to minimize the total increased amount of inconvenience to passengers, which is a valid extension of passengers' arrival delay, over all origin-destination pairs of the passengers.

We present rolling stock rescheduling of passenger or freight trains based on (K.) Sato and Fukumura (2012). In the railway industry, our rolling stock rescheduling is the first approach that formulates the rescheduling as an integer programming problem which can be considered as a variant of the set partitioning problem. We apply column generation, and temporarily introduce relaxation of the original formulation named set-covering relaxation, to solve the problem in acceptable time. In the column generation of our algorithm, a feasible schedule of each rolling stock unit is generated in polynomial time. In such a schedule, the necessary number of inspections is carried out at a proper interval and the number can be two or more when a rescheduling period is long. Our algorithm does not necessarily provide an optimal rolling stock rescheduling solution for every instance due to a limited number of columns to be generated. It provides an optimal solution or decides the infeasibility of the instance correctly, however, when the column generation terminates with an integral solution.

We discuss crew rescheduling of passenger or freight trains based on (K.) Sato and Fukumura (2011a). In the railway industry, our crew rescheduling is one of the first approaches that formulate the rescheduling as an integer programming problem which can be considered as a variant of the set covering or the set partitioning problem. The problem is solved by column generation. Our algorithm does not necessarily provide an optimal crew rescheduling solution for every instance due to a limited number of columns to be generated. It provides an optimal solution or decides the infeasibility of the instance correctly, however, when the column generation terminates with an integral solution.

1.4 Outline

The remainder of this thesis is organized as follows. Chapter 2 gives an overview of rail transport rescheduling. The next three chapters provide integer programming approaches to the rail transport rescheduling. In Chapter 3, we discuss timetable rescheduling. We then present models and algorithms for rolling stock rescheduling in Chapter 4. In Chapter 5, we deal with crew rescheduling. Chapter 6 concludes our work. We lastly state future work and prospects for practical real-time rescheduling.

Chapter 2

Rail Transport Scheduling and Rescheduling

2.1 Rail Transport Scheduling

2.1.1 Classification of Problems

Rail transport scheduling has been one of major applications of operations research and mathematical optimization. Hillier and Lieberman (2014), Nemhauser and Wolsey (1999) and Wolsey (1998) expound various theoretical topics on these fields. Practical approaches based on theoretical work have brought a great success to some train operating organizations. Ben-Khedher et al. (1998), Ireland et al. (2004) and Kroon et al. (2009) are monumental work.

Caprara et al. (2007) discusses passenger railway scheduling, mainly focusing on that in Europe. Six scheduling problems are presented: line planning, train timetabling (we call it timetable scheduling in this thesis), train platforming, rolling stock circulation (rolling stock scheduling), train unit shunting (shunting scheduling) and crew planning (crew scheduling), which are planned in this order. The line planning is to decide a route and stop stations of train services on a railway network. The timetable scheduling is to provide arrival and departure times of all the trains at all the predetermined points on their route, whose typical example is a station. A route of each train inside every station is decided in the train platforming, including a track which is adjacent to a platform and which the train passes through or stops at. The rolling stock scheduling is assignment of rolling stock units to each of the timetable. Shunting is applied to rolling stock units and is carried out at rolling stock depots and certain stations, whose processes include their routing as well as their coupling and uncoupling. They are planned in the shunting scheduling. A train crew is composed of mostly one driver and one or a few conductors. The crew scheduling is assignment of them to each of the timetable. The scheduling is further classified into short-term, typically one or two days, planning of work, and long-term one in which the short-term plans are combined. The latter is called crew rostering.

According to Planning Systems Laboratory, Railway Technical Research Institute (2005), rail transport scheduling is classified into four units in Japan: timetable scheduling, rolling stock scheduling, crew scheduling and shunting scheduling. The rolling stock, the crew and the shunting scheduling depend on a planned timetable. The latter schedules depend on the former schedules. For instance, crew members have been trained to operate limited types of rolling stock unit. Shunting operations in a rolling stock depot or a station are based on the arrivals and departures of rolling stock units there. On a certain line, drivers who operate trains between stations also do inside a rolling stock depot or a station. The scale and complexity of each scheduling phase makes all of them difficult to be handled simultaneously. Therefore, they are planned in the first described order on the whole, except that the rest of the schedules can be anticipated, for instance, from the current ones, and that a part of the work can be carried out independently. In this country, a timetable is mostly planned for each railway line and trains running on the line pass through or stop at one of a few predetermined tracks at a station. At some stations, only one track exists for each direction towards which trains run. Trains are ordinarily classified into a few or several types and stop stations on the line are decided for each type. Line planning and train platforming are hence integrated into the timetable scheduling. Arrival and departure times of a train which runs through two or more lines are often discussed between timetable planners of the involved lines, or an integrated timetable of trains which run on the lines is scheduled. When trains temporarily occupy the same piece of infrastructure on an ordinary railway line, their orders should be planned. The departure order of two trains from a station which will run toward the same railway track segment is a typical example. In a rolling stock schedule, the position of a rolling stock unit should also be stated when a train is composed of two or more units. A periodic inspection (or called maintenance) of each unit, for every a few or several days, has to be carried out. It takes a few hours. It is also specified when and where every inspection is done. In a crew schedule, work of any crew member for one or two days ordinarily begins and ends at the same crew base to which he/she belongs to.

2.1.2 Scheduling Entities

On many European railway lines and networks, a single infrastructure manager and train operators are the organizations involved in rail transport, according to Caprara et al.

(2007). The infrastructure manager is responsible for train planning and real-time traffic control, whereas the train operators provide transport services. The operators can be split into operators of passenger trains and operators of cargo trains. The train operators perform line planning and make an ideal timetable of their train services. The infrastructure manager deals with timetable scheduling and train platforming, based on the requests by the train operators. Rolling stock scheduling, shunting scheduling and crew scheduling are planned by the train operators.

In Japan, the organizations involved in rail transport are typically divided into passenger railway operators and freight train operators. Each of most railway lines is operated by a single passenger railway operator, who also owns the infrastructure. Meanwhile, most freight trains operated by a freight train operator run on a network which consists of several lines, and infrastructure on each of them is managed by a passenger railway operator. The timetable, the rolling stock, the crew and the shunting scheduling are all planned by each operator. The timetables of the freight trains and ones which run through several lines owned by different operators are then discussed and adjusted by timetable planners of the relevant operators.

2.1.3 Scheduling Approaches

The readers are recommended referring to Assad (1980), Bussieck et al. (1997), Caprara et al. (2007), Cordeau et al. (1998), Crainic and Laporte (1997), Huisman et al. (2005), Narayanaswami and Rangaraj (2011) and Planning Systems Laboratory, Railway Technical Research Institute (2005) on the 60-year history of operations research and mathematical optimization approaches to rail transport scheduling. Planning Systems Laboratory, Railway Technical Research Institute (2005) presents approaches to scheduling problems in Japan. Timetable scheduling is the most vigorously discussed topic among all the scheduling phases. ON-TIME (2013) performs an analysis and assessment of methods for timetable scheduling. Various topics related to a railway timetable are presented by Hansen and Pachl (2014). Aronsson et al. (2009) and Harrod (2012) present models for timetable scheduling.

Robust schedules against disturbance to railway operations are also studied. Cacchiani and Toth (2012) presents a review of robust timetable scheduling. De Almeida et al. (2008) discusses robust rolling stock scheduling and Cadarso and Marín (2011) robust rolling stock scheduling and shunting. Robust crew scheduling is discussed by Flier et al. (2007). The concept of “recoverable robustness” is introduced and applied to timetable scheduling by Liebchen et al. (2009). A schedule is called recovery robust if it can be rescheduled by limited rescheduling measures in all likely disruption scenarios. The multi-stage version of the recoverable robustness is presented by Cicerone et al. (2012). Caprara et al. (2014) also extends the original concept. Cacchiani et al. (2012) provides a similar concept in rolling stock scheduling. Recovery robust shunting scheduling is discussed by Cicerone et al. (2009a) and that in timetable and shunting scheduling is done by Cicerone et al. (2009b).

2.2 Rail Transport Rescheduling

2.2.1 Rescheduling Process and Entities

Trains on a railway line or a railway network is operated in accordance with a set of schedules planned in the rail transport scheduling. A train traffic control system is ordinarily installed to manage the train operations. These railway operations are occasionally disrupted by disturbance. Rail transport rescheduling is carried out, at a certain point of time, when the railway operations in accordance with the current set of schedules are being, or those to be implemented from the time to the following hours will be, delayed to some extent or unable to be carried out, owing to the disturbance.

Sources of the disturbance include, accidents, bad weather conditions or natural disasters, malfunctions of components of rolling stock units as well as railway facilities such as traffic signals and switches, and trouble to passengers as well as crew members. Even many passengers' boarding and alighting from a train one after another can be a source of disturbance, if the required dwell time exceeds the planned one. Loading and unloading of cargos can also be sources of disturbance.

As we have described, the timetable scheduling, if we regard line planning and train platforming as a part of it, is first of all planned. It is then followed by the rolling stock, the crew and the shunting scheduling in this order. In the greater part of Japan and the world, the rail transport rescheduling is designed and implemented in the same order, if all of them are required. One reason for this is that the other rescheduling depends on the rescheduled timetable. Another is that their scale and complexity would make the problems difficult to be handled simultaneously. The crew and the shunting rescheduling are followed by the rolling stock rescheduling, since a rescheduled rolling stock plan might affect them. If the rolling stock is rescheduled, there will be a possibility that a certain driver might have never been trained to operate a reassigned type of rolling stock unit and will not be permitted to drive it. The shunting rescheduling is independent of the crew rescheduling, except for the case where the same driver operates trains between stations and inside a rolling stock depot or a station. Studies of the problems contained in the whole rescheduling process are reviewed by Jespersen-Groth et al. (2009a), Kroon and

Huisman (2011) as well as Narayanaswami and Rangaraj (2011).

A staff in charge of the rail transport rescheduling inside railway organizations is called dispatchers. On many European railway lines and networks there are two or more organizations involved in the rail transport, as we have stated. A single infrastructure manager deals with the timetable rescheduling, whereas train operators which provide transport services on the infrastructure do the rest of the rescheduling. Even though it is the case that the whole rescheduling is handled inside a single railway operator, which is true for most passenger railway lines in Japan, each of the rescheduling is implemented by different dispatchers. As we have described, most freight trains operated by a freight train operator in Japan run on a network which consists of several lines, and infrastructure on each of them is managed by a passenger railway operator. The passenger railway operators reschedule the freight train timetable and the freight train operator does the rest of the rescheduling. The freight train operator can delegate the rolling stock and the driver rescheduling to the passenger railway operators. The series of rescheduling is ordinarily designed in one or more train traffic control centers except for the shunting rescheduling. It is designed and carried out at a rolling stock depot or a station if required there.

The rescheduling plans have to be designed and implemented in real-time, since the initial disruption of trains by disturbance is being spread to other trains, except for small disruption cases where they are resolved by buffers in the timetable. Another case is that a rolling stock unit or a crew member has not yet been assigned to trains which will depart after the following minutes. There is also a case where the disrupted situation might change while we are considering rescheduling plans and where the plans might not be applicable in the latest situation. There is hence a deadline for planning a new schedule in each of the rescheduled phases. After that, the dispatchers have to input the modified pieces from the current schedule in the new schedule into a relevant train traffic control system or communicate it to a relevant staff. Therefore, rescheduling start time is set and the schedule from the time onward is designed.

2.2.2 Timetable Rescheduling

The dispatchers in charge of the timetable rescheduling perform a series of actions which change a timetable to maintain the level of the transport services in the situation. They mainly focus on train delays and try to reduce them. Some of the trains are canceled when the disruption is large. On the other hand, direct impacts of the rescheduling on the passengers have also been considered. One example is to keep a connection between trains running on a different line at a station where the two trains stop. Even if a train on one line is delayed by disturbance, a train on the other line will wait for the former so that the passengers can transfer between them. Another is to observe or even predict the appearance of passengers at a station and offer train operations to minimize their waiting time. These kinds of rescheduling actions are receiving more and more attention of both the dispatchers and the passengers, and therefore we have to make the rescheduled plan rather passenger-oriented than train-punctuality-oriented.

As we have discussed above, there are various sources of disturbance and hence its scale varies from case to case. When the disruption is large, rule-based general rescheduling measures are often applied. We discuss the situations where the disruption is small or the general rescheduling has already been implemented. In such situations, all of the trains are due to be and can be operated between the ends of the whole line or a subset of the line divided by the disturbance. We focus on detailed rescheduling in which we have to deal with arrival and departure times of each train at each station to manage its delay. We assume that no out-of-service train is operated.

We let an originally planned timetable, the current timetable, initial delays of trains, a set of trains to be rescheduled and the number of passengers for every station pair and time period (passenger OD data) be given on a passenger railway line. The timetable rescheduling in this thesis is then to manage their delays by applying a series of detailed rescheduling measures so that inconvenience to passengers caused by the disturbance should be minimized. Various kinds of constraints on the train operations has to be satisfied so that the trains could be operated in accordance with the rescheduled timetable. A series of constraints on the passenger's behavior has also to be satisfied so that his/her route should be rational. We have to create a new timetable from the rescheduling start time before the rescheduling design deadline comes.

2.2.3 Rolling Stock Rescheduling

As we have briefly mentioned, the dispatchers in charge of the rolling stock rescheduling have to communicate the modified pieces from the current schedule in the new schedule to train crews and a rolling stock depot staff, via a station staff, by radio, by phone or by any information technology. This task is cumbersome, hence the scale of the actions to be rescheduled should be as small as possible. Another difficulty in the rolling stock rescheduling is the management of the periodic inspections. We have to ensure that inspections are also properly carried out in a redesigned schedule of each rolling stock unit. An unscheduled inspection can be carried out, though it is not desirable since it will be a burden on the rolling stock depot staff and the number of units that we can simultaneously inspect is limited at each depot and at any point of time. We have to make the rescheduled plan operator-oriented by taking these requests into consideration.

We let an updated timetable, the current rolling stock schedule and the disrupted situation be given on a railway line or a railway network of passenger or freight trains. The rolling stock rescheduling in this thesis is then to reassign the rolling stock units the trains so that the scale of the actions to be rescheduled should be as small as possible and that unscheduled inspections should be as few as possible. Two consecutive inspections of each of the units must not violate the maximum inspection interval permitted. Each train is assumed to consist of one unit, which is common on many lines and networks, or to be composed of two or more units which can be freely coupled. Routing inside depots and stations as well as coupling and uncoupling of units are discussed in the shunting rescheduling. A deadhead train can be canceled or up to the specified number of coupled units. A rolling stock unit is not permitted to be assigned to a train if its type is not suitable for the characteristics of the train. Preparation time is necessary between two consecutive train operations to which the same unit is assigned. We have to reschedule the rolling stock assignment to the trains involved to be operated from the rescheduling start time to hours after that before the rescheduling design deadline comes.

2.2.4 Crew Rescheduling

As we have briefly mentioned, the dispatchers in charge of the crew rescheduling have to communicate the modified pieces from the current schedule in the new schedule to the crews via a station or crew base staff, by radio, by phone or by any information technology. This task is cumbersome, hence the scale of the actions to be rescheduled should be as small as possible. Meanwhile, some crew members are obliged to work overtime by their new schedule, which is not desirable. We have to make the rescheduled plan operator-oriented by taking these requests into consideration.

We let an updated timetable, an updated rolling stock schedule, the current crew schedule and the disrupted situation be given on a railway line or a railway network of passenger or freight trains. The crew rescheduling in this thesis is then to reassign the crew members the trains so that the scale of the actions to be rescheduled should be as small as possible and that overtime should be as short as possible. In the rescheduling, we have to let each of the members return to the crew base to which he/she belongs to. The number of crew members essential to each train depends on the characteristics of the train. Ordinarily, the deadhead trains require one driver and no conductor, whereas the other trains require one driver and one or more conductors. A crew member is not permitted to operate a train if he/she has never been trained to work in the railway section where the train runs or to operate the rolling stock unit assigned to the train. Preparation time is necessary between two consecutive train operations by the same crew member. We have to reschedule the crew assignment to the trains to be operated from the rescheduling start time to hours after that before the rescheduling design deadline comes.

2.2.5 Shunting Rescheduling

The shunting rescheduling is designed and carried out at each rolling stock depot or each station if required there. The problem is defined as follows. An updated timetable and a rolling stock schedules, as well as a crew schedule if they are involved in the shunting, and the current shunting schedule are given. This task is then to reschedule the routing of the rolling stock units inside the facility, their coupling and uncoupling and other necessarily operations, so that the departure time and the rolling stock unit composition of each train have to correspond to them specified in the rescheduled timetable and rolling stock plan, respectively. Trains which run through the facility must not be affected by the unrescheduled and rescheduled shunting operations. The scale of the actions to be rescheduled should be as small as possible and the operations from the rescheduling start time to hours from that have to be redesigned before the rescheduling design deadline comes, as is the case with the rolling stock and the crew rescheduling.

Budai et al. (2010), Jespersen-Groth et al. (2009b), Kroon et al. (2015), Nielsen (2011, Chapter 3) and Nielsen et al. (2012) as well as Tomiyama et al. (2012a) take coupling and uncoupling of rolling stock units into account in rolling stock rescheduling, whereas routing inside depots or stations is omitted. Cadarso et al. (2013) integrates timetable rescheduling into these two rescheduling. The approaches by Fukumura and Maruyama (2008) and Sugi et al. (2010) are dedicated to the shunting rescheduling in Japan. The shunting scheduling algorithm proposed by (T.) Sato et al. (2007) for a rolling stock depot in Japan is also applied to the rescheduling. Demange et al. (2012) views shunting as an online optimization problem (refer to Borodin and El-Yaniv (2005) on online optimization). Winter and Zimmermann (2000) discusses both the offline and the online cases.

Chapter 3

Timetable Rescheduling

3.1 Introduction

3.1.1 Background

This chapter provides timetable rescheduling of passenger trains while railway operations on a line are disrupted by disturbance, based on (K.) Sato et al. (2013). A planned timetable of train operations specifies, for each train, its route, stations which it passes through or stops at along the route and arrival time and departure time at each station. The route is composed of track segments between each pair of stations on which the train runs and those inside each station. Inside a station at which the train stops and passengers board and alight, the track at which it stops is important to the passengers. The type of the train represents the sets of the stations which it passes through and stops at. Out-of-service trains are operated between a rolling stock depot and a station, or even between stations, related to a rolling stock schedule. When trains temporarily occupy the same piece of infrastructure on an ordinary railway line, their orders should be planned. The arrival and departure orders of the trains at the stations which run on or stop at the same railway track segment are typical examples. A train traffic control system is originally installed to manage the train operations in accordance with the timetable. The timetable rescheduling is carried out, at a certain point of time, when the railway operations in accordance with the current timetable are being, or those to be implemented from the time to the following hours will be, delayed to some extent or unable to be carried out, owing to the disturbance.

Among the rail transport schedules, the timetable is most subject to disturbance. Disturbance which primarily affects the rolling stock schedule, a malfunction of a component of a rolling stock unit for instance, influences train operations according to the timetable, if the rolling stock is not rescheduled and recovered by the time when the operations are carried out. Disturbance which affects the crew schedule, injury of a crew member on his/her at work for instance, or which affects the shunting scheduling, may also lead to the disruption of the train operations.

The timetable rescheduling plan has to be designed and implemented in real-time, since the initial disruption of trains by disturbance is being spread to other trains, except for small disruption cases where they are resolved by buffers in the timetable. Another case is that the disrupted situation might change while we are considering a rescheduling plan and that the plan might not be applicable in the latest situation. The dispatchers in charge of the timetable rescheduling perform a series of actions which change a timetable to maintain the level of the transport services in the situation. They mainly focus on train delays and try to reduce them. Some of the trains are canceled when the disruption is large. On the other hand, direct impacts of the rescheduling on the passengers have also been considered. One example is to keep a connection between trains running on a different line at a station where the two trains stop. Even if a train on one line is delayed by disturbance, a train on the other line will wait for the former so that the passengers can transfer between them. Another is to observe or even predict the appearance of passengers at a station and offer train operations to minimize their waiting time. These kinds of rescheduling actions are receiving more and more attention of both the dispatchers and the passengers, and therefore we have to make the rescheduled plan rather passenger-oriented than train-punctuality-oriented.

3.1.2 Terminology

On the basis of the disruption management context described above, we define terminology of the timetable rescheduling on a railway line as follows. We call a delay of a train directly and primarily caused by disturbance an *initial delay*. Disturbance may affect more than one train at a time. An initial delay includes a forecast of the amount of the delay, based on historical data on the same type of disturbance for instance. We call a certain point of time at which we have noticed necessity for the rescheduling the *current time*. A certain amount of time is required to plan a new schedule, and input the modified pieces from the current schedule in the new schedule to the train traffic control system installed on the line and communicate it to a relevant staff such as a rolling stock depot staff, train crews and a station staff, if necessary. We let a *rescheduling design deadline* be a deadline for planning the new schedule. The time when the input and communication is completed and we can implement the schedule is called *rescheduling start time*.

An *originally planned timetable* denotes a published nominal timetable after thoroughly planned in the timetable scheduling. The *current timetable* is the originally planned timetable if no rescheduling has yet been carried out, of the latest timetable if the originally planned timetable has been rescheduled once or more in any manner. Arrival and departure time of trains in a timetable is denoted by *train operations*.

A *rescheduling measure* is an action which changes the current timetable. Such actions are classified into two types: *general* and *detailed* rescheduling measures. General rescheduling measures are applied when the disruption is large and they include train cancellation, global rerouting of a train which is also called a detour to avoid the disrupted area, and shuttle operations between undisrupted areas. Without these rescheduling measures, a larger number of trains will be involved in the disruption. Detailed rescheduling measures include *reordering* of trains, *local rerouting* of a train inside a station, *retiming* of a train, *change of a train type* and *change of rolling stock assignment* to a train. Their examples are presented in a later section.

A pair of passenger's origin station and his/her destination station is commonly called *OD pair*. We call *passenger OD data* the number of passengers for every OD pair who appear at their origin station at every time period. A *passenger's route* is defined as a sequence of stations and trains which he/she takes and transfers. A choice of a passenger's route is denoted by *passenger's behavior*. A passenger will suffer *inconvenience* of traveling by train which is operated according to an originally planned timetable or a rescheduled timetable. The inconvenience can be seen as his/her traveling time typically. He/she would suffer no inconvenience if he/she could instantly travel from his/her origin station to destination. Other factors will include waiting time at a platform, train congestion and a train transfer. He/she may change his/her behavior when a rescheduled timetable is put into action and announced. We call it *passenger rerouting*.

3.1.3 Approach

As we have discussed above, there are various sources of disturbance and hence its scale varies from case to case. We focus on the time span and the space areas which corresponds to the shaded area of the time-space diagrams displayed in Figures 3.1 and 3.2. The first case is that the disruption is large and general rescheduling measures have been applied. The left of Figure 3.1 shows the originally planned timetable and the middle displays disturbance with no rescheduling measure being performed. The right of this figure shows that the shuttle operations between Station A and Station B as well as between Station C and Station D are performed. The other case is shown in Figure 3.2; the disruption is small and no general rescheduling measure is required. We discuss the situations where all of the trains are due to be and can be operated between the ends of the whole line or a divided subset of the line, and require detailed rescheduling measures to manage the

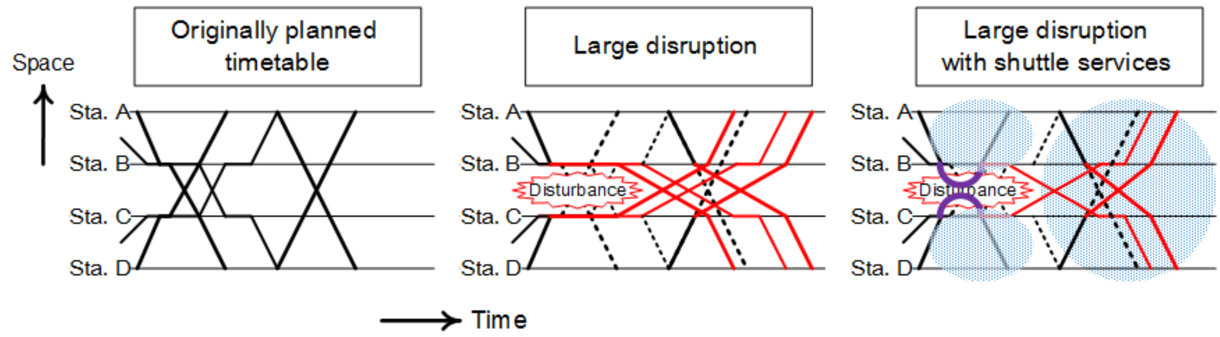

Fig. 3.1 Large disruption

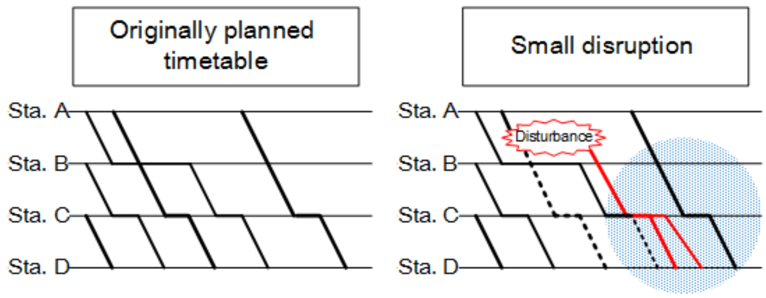

Fig. 3.2 Small disruption

delay. We assume that the passengers can board every train.

We let an originally planned timetable, the current timetable, initial delays, a set of trains to be rescheduled, passenger OD data, the rescheduling design deadline and the rescheduling start time be given on a passenger railway line. The *timetable rescheduling* in this chapter is then to manage their delays by applying a series of detailed rescheduling measures so that inconvenience to the passengers caused by the disturbance should be minimized. Various kinds of constraints on the train operations has to be satisfied so that the trains could be operated in accordance with the rescheduled timetable. A series of constraints on the passenger's behavior has also to be satisfied so that his/her route should be rational. We have to create a new timetable from the rescheduling start time before the rescheduling design deadline comes.

We formulate the timetable rescheduling problem as mixed integer programming problems. We let events in the train operations be arrivals or departures of trains, and change their event time. Among detailed rescheduling measures, reordering, local rerouting and retiming of trains are formulated. We also apply change of a train type and change of rolling stock assignment. We let passenger rerouting be possible and the passengers may transfer once or more on their route. The train operations and the passengers' behavior are simultaneously modeled. We also let the inconvenience to each passenger consist of his/her traveling time on board, waiting time at platforms and the number of transfers. We first model and solve the arrival delay minimization problem. We next introduce some flexibility in the delay-minimized timetable and solve the passenger inconvenience minimization problem. The readers are recommended referring to Hillier and Lieberman (2014), Nemhauser and Wolsey (1999), Williams (2013) and Wolsey (1998) on mixed integer programming. Other topics on operations research and mathematical optimization are also expounded by Hillier and Lieberman (2014), Nemhauser and Wolsey (1999) and Wolsey (1998).

3.2 Literature Review

3.2.1 Classification of Approaches

Before reviewing disruption management research in the railway industry, we briefly mention that in other transportation industries. Timetable rescheduling in the airline industry, in conjunction with aircraft rescheduling, has been vigorously discussed. Clausen et al. (2010) provides an extensive review in the industry. An overview of disruption management in the maritime industry is presented by Qi (2015). Brouer et al. (2013) and Li et al. (2015) study vessel rescheduling, which include timetable rescheduling. General vehicle rescheduling on a road network is reviewed by Visentini et al. (2014). In some models, delay is tried to be minimized. As Clausen (2007) points out, rolling stock runs on a one-dimensional railway, and this makes railway timetable rescheduling distinct from the rescheduling in the other industries.

There has also been a large number of timetable rescheduling studies in the rail transport rescheduling literature. Timetable rescheduling approaches depend on the scale and the situation of disruption. Technical Research Committee on Advanced Rail Transport Planning and Management, The Institute of Electrical Engineers of Japan (2010) presents various rescheduling measures and disrupted situations where each of them is applied. Based on this work, we have distinguished between general and detailed rescheduling measures in the preceding section. Recall that the former measures are first implemented when the disruption is large. The latter is performed when the disruption is small or certain rescheduling measures have already been implemented.

General rescheduling measures adjust frequency of train services on the whole, while detailed rescheduling measures are applied in order for delay of the trains to be managed. We classify approaches to handle the delay into two types: train-punctuality-oriented and passenger-oriented ones. The train-punctuality-oriented approaches simply minimize the amount of delays, the number of trains to be delayed or variants of these. A weight is sometimes set for each train and it can be the number of passengers which is assumed to be on board. We do not regard, nevertheless, this type of model as a passenger-oriented one. We define the passenger-oriented approaches as rescheduling models which include behavior of passengers which depend on a rescheduled timetable and optimize a criterion related to the behavior such as waiting time of the passengers at their original stations and their arrival delay at their destination stations.

Spatial targets for the rescheduling on a network also vary from research to research. We classify them into four categories. One is a station on a line which has many tracks or sidings or a junction where two or more lines are connected. The rest of the categories are, an area which includes a few or several stations, the whole or a partial line and the whole or a partial network.

In the rail transport scheduling and rescheduling literature, rescheduling models are also categorized according to their granularity. A train movement on each track segment or occasionally inside a segment of a railway line or a railway network is modeled in microscopic models. On the other hand, each station or a key station which connects lines, at which many passengers board or alight from a train, etc., is ordinarily a minimum unit of modeling in macroscopic models.

3.2.2 Train Stop Deployment Planning and Emergency Timetable Application

Corman and D'Ariano (2012) presents an assessment of "emergency timetables," which are prepared in advance, and one among them is implemented as a response to large disruption in the Netherlands. An example of emergency timetables is a rescheduled one in which general rescheduling measures are combined such as train cancellation, global rerouting of a train to avoid the disrupted area and shuttle operations in the disrupted area to keep the service level there when the section is not being completely blocked by the disturbance. Based on a similar concept, "rescheduling patterns" are prepared in Japan. Hirai et al. (2006) and Nakamura et al. (2011) discuss implementation of the rescheduling patterns. In their models, shuttle operations are performed in the undisrupted area when the disrupted section is being completely blocked by the disturbance. Hirai et al. (2009) and Hirai and Takahashi (2009) introduce a "train stop deployment planning" problem, which is discussed before we apply an emergency timetable or a rescheduling pattern. This problem decides stop locations of disrupted trains, sidings at stations for instance, not to interfere with train operations which run through the undisrupted area. Louwerse and Huisman (2014) takes an optimization approach in the large disruption. Jespersen-Groth and Clausen (2006) discusses reinsertion of trains after the disturbance which forced train cancellation on the whole line has ended.

3.2.3 Train-punctuality-oriented Rescheduling

The majority of the timetable rescheduling studies take train-punctuality-oriented approaches. Such studies presented before around the middle of the 2000s are reviewed by Cordeau et al. (1998), Goodman and Takagi (2004) and Törnquist (2006).

In Japan, according to Akita and Hasegawa (2013), train reordering and track reassignment algorithms at a station were implemented in the traffic control system of trains on Tokaido and Sanyo Shinkansen lines (high-speed trains in Japan) called COMTRAC in 1972. It is based on a macroscopic model, while most studies and practical applications are microscopic. On Chuo Line in Tokyo, which is one of railway lines called conventional lines meaning that a Shinkansen train are not operated on the line, another train reordering algorithm is implemented in the traffic control system called ATOS by Kitahara et al. (1998). Schaafsma and Bartholomeus (2007) presents an algorithm introduced around Schiphol station, the Netherlands, and Mannino and Mascis (2009) at metro stations in Milan, Italy. Other rescheduling studies at a station or a junction include Chen et al. (2010), Lusby et al. (2013), Pellegrini et al. (2014) and Rodriguez (2007). Flamini and Pacciarelli (2008) considers headway between trains into account. Energy consumption of trains is added to the optimization criterion by Mascis et al. (2008). Fan et al. (2012) assesses performance of several approaches. Caprara et al. (2010) views timetable rescheduling at a station as an online optimization problem (refer to Borodin and El-Yaniv (2005) on online optimization).

Train-punctuality-oriented rescheduling in an area which includes a few or several stations are often formulated as microscopic models. Above all, the “alternative graph” formulation by D’Ariano et al. (2007a), which is based on the research on a variant of the job-shop scheduling problem by Mascis and Pacciarelli (2002), is widely used in many succeeding studies. Espinosa-Aranda and García-Ródenas (2013), Khosravi et al. (2012) and Liu and Kozan (2009) as well as Mladenović and Čangalović (2007) also deal with this problem. Boccia et al. (2013) adds use of an unpreferred track at a station by each train to the objective function. D’Ariano et al. (2007b) also controls train speed. Local rerouting inside a station is taken into account by Corman et al. (2010c) and Corman et al. (2011a) as well as D’Ariano et al. (2008b). The last one also assesses the amount of delay if and if not connections between trains are maintained. Corman et al. (2009a) takes train energy consumption into account. Mazzarello and Ottaviani (2007) discusses global rerouting in addition. Corman et al. (2011b) reschedules trains in accordance with their priority. A long rescheduling time horizon is decided into tractable intervals and the rescheduling in each time span is solved by D’Ariano and Pranzo (2009). Corman et al.

(2010a) presents and a distributed rescheduling model compares the previously developed centralized model with it. Corman et al. (2009b) compares two models: one based on track segments and the other based on the aggregation of them into station routes. Corman et al. (2010b) and D'Ariano et al. (2008a) assess performance of different approaches.

Detailed timetable rescheduling on the whole or even a partial line is a large-scale problem, and macroscopic models are hence presented in most studies. This topic is vigorously discussed in Japan. Major studies are by Abe and Araya (1985), Araki et al. (1995), Araya and Abe (1983), Sakikawa et al. (2010) and Tomii and Ikeda (1995). Komaya and Fukuda (1991) presents a rescheduling support system called ESTRAC-III, which is applied to Tokaido Shinkansen. Shimizu et al. (2001) and Shimizu et al. (2008) presents algorithms implemented in the traffic control system called COSMOS to manage the traffic of trains on Tohoku, Akita, Yamagata, Joetsu and Hokuriku Shinkansen lines. In other countries, Afonso and Bispo (2011), Dündar and Şahin (2013), Fekete et al. (2011) and (X.) Meng et al. (2010) discuss this topic. Walker et al. (2005) deals with timetable and crew rescheduling simultaneously. The algorithm proposed by Lamorgese and Mannino (2015) has been in operation on the main line in the Stavanger region, Norway. An online optimization approach is adopted by (L.) Meng and Zhou (2011).

Kecman et al. (2013), Min et al. (2011), Törnquist and Persson (2005), Törnquist (2007), Törnquist and Persson (2007) and Törnquist Krasemann (2012) discuss train-punctuality-oriented rescheduling on a network. They are based on macroscopic models. Connections between trains are formulated as a constraint by Acuna-Agost et al. (2011a) and Acuna-Agost et al. (2011b). Iqbal et al. (2013) proposes a parallelized multi-strategy based algorithm. Time events of each train is computed according to four strategies concurrently and the best one is chosen. Note that there are microscopic approaches to timetable rescheduling on a network. Corman et al. (2012b) and Corman et al. (2014) divide a network into several areas, solve a rescheduling problem for each area and coordinate the outputs. Luethi et al. (2009) presents an online train rescheduling system on a network in Switzerland.

3.2.4 Passenger-oriented Rescheduling

To make detailed rescheduling more passenger-oriented, we have to focus on passenger OD pairs. Therefore, a spatial target is inevitably a line or a network. As we have reviewed above, macroscopic models are ordinarily presented in a rescheduling problem on a line or a network. A passenger-oriented quantitative rescheduling criterion was proposed nearly 40 years ago in Japan. Nagai et al. (1977) compares four rescheduling strategies and evaluate them for an imaginary rescheduling case of trains on Tokaido and Sanyo Shinkansen lines in terms of missed seat reservations, arrival delay of passengers, increase traveling time and train congestion.

Research on passenger-oriented rescheduling algorithms emerged several years later. Araya and Sone (1984) discusses controls of trains on a simple railway loop whose track layout is a single track. The control policy is to minimize passengers' traveling time and prevent their discomfort due to train congestion. A fuzzy expert system approach is adopted by (C. S.) Chang and Thia (1996). A multiobjective evolutionary algorithms is proposed by Kwan and Chang (2005).

Goverde (1998) formulates delay propagation of train operations in accordance with a cyclic timetable as a max-plus linear system and minimizes waiting time of passengers at transfer stations. A timetable called cyclic trains whose route and stop stations are the same periodically come and go, for each 30 minutes, one hour, etc. Similar models are presented by de Vries et al. (1998) and De Schutter et al. (2002). In their models, both total delay of trains and connections to be kept are taken into account. A train reordering option is added by van den Boom and De Schutter (2006). Different models are presented by van den Boom and De Schutter (2007) as well as Kersbergen et al. (2013), and they are converted into mixed integer programming problems.

Adenso-Díaz et al. (1999) maximizes the number of passengers to be transported. The number of passengers who want to board each train is given, and a train will be canceled if it exceeds the maximum permissible delay at the departure. Computational experiments are performed with data of regional railway network in Asturias, Spain. The sum of various terms, including the average passenger travel time, is minimized by (S.-C.) Chang and Chung (2005). Instances of trains in Taiwan are solved by a genetic algorithm. Wegele and Schnieder (2005) also presents a genetic algorithm to minimize an objective function which includes a penalty of breaking a connection. Numerical experiments are carried out on practical data delivered by Deutsche Bahn AG in Germany. Boland et al. (2012) provides a mixed integer programming formulation in which boarding time of a train at a station is lengthened when the train is crowded. Trains can skip stations. Traveling time of passengers on Sandringham Line in Australia are minimized, though it takes time. Corman et al. (2012a) presents a bicriteria problems minimizing the total delay and the weighted number of connections to be kept. Note that the presented model is microscopic and the dispatching area around Utrecht Central Station in the Netherlands is discusses.

Eberlein et al. (1999) minimizes the total passenger waiting time on a simple loop. Skipping a station by either making the train out of service or operating it as an express train is allowed. Heuristics is proposed and applied to train operations on MBTA Green Line B in Boston, the U.S. O'Dell and Wilson (1999) presents a mixed integer programming model and minimizes the total passenger waiting time or the total passengers' traveling time.

Train reordering and the capacity of trains are taken into account. Optimal solutions for many instances train operations on MBTA Red Line in Boston, the U.S., is obtained in 30 seconds. Shen and Wilson (2001) presents another mixed integer programming model and compares several strategies for each instance of trains operations on the Red Line.

Suhl and Mellouli (1999) discusses a decision support system for dispatchers of Deutsche Bahn AG in Germany. Suhl et al. (2001b) presents the simulation part of the systems and compare four strategies concerning a connection of trains by utilizing the simulator. Suhl et al. (2001a) also presents optimization models which decides whether or not each connection is to be kept, so that the weighted waiting time of passengers including the wait time for a connection should be minimized. The instances are solved within 200 seconds. Biederbick and Suhl (2007) compares 14 rescheduling strategies.

Cadarso et al. (2013) proposes simultaneous rescheduling of the timetable and the rolling stock plans based on the scheduling algorithms proposed by Cadarso and Marín (2011) as well as Cadarso and Marín (2012). Coupling and uncoupling of the units are also considered. The objective function includes the number of passengers which cannot board due to the limited capacity of the train composition. The train rescheduling and the passengers' reaction to the rescheduled timetable are also iteratively solved. These algorithms are applied to instances of train operations on a network in Madrid, Spain. Veelenturf et al. (2011) integrates rescheduling of rolling stock and the adaptation of stopping patterns and provides heuristics.

Yang et al. (2002) minimizes the sum of increased amount of passengers' traveling time and train congestion and allows change of stop stations. Simulated annealing is applied to this problem. Goodman and Murata (2001) minimizes the weighted sum of traveling time, waiting time, train congestion and energy consumption of trains on the Hong Kong Island Line by a gradient-based search procedure. Nagasaki et al. (2003) takes a similar approach, and change of a train type from a low one to a high one as well as train reordering are applied to trains on Shinjuku Line operated by SEIBU Railway in Japan. Takagi et al. (2004) compares two strategies concerning connections around Highbury & Islington Station, the U.K.

Tomii et al. (2005) defines a priori several kinds of situations where passengers might complain, which are arrival and departure delays of trains over a certain threshold, low frequency of train services, a break of a connection, etc., and tries to minimize the weighted number of complaints by simulated annealing. Based on a network proposed by Abe and Araya (1986) which expresses train operations, various general and detailed rescheduling measures are applied, including change of rolling stock assignment. Hence this approach makes no assumption on the scale of the disruption. Kunimatsu and Hirai (2009) partially adopts this algorithm and takes an iterative approach which simulates detailed passengers' behavior in a tentative rescheduled timetable and changes the timetable with the passengers' behavior being fixed. The optimization criterion is the disutility value proposed by Railway Bureau, Ministry of Land, Infrastructure, Transport and Tourism (2012), which is commonly used by civil engineers in Japan to evaluate rail projects. The disutility value of each passenger is composed of his/her traveling time on board, waiting time at platforms, the number of transfers and a piecewise linear convex and increasing function value of train congestion. The computation time is not mentioned. A similar iterative algorithm is proposed by Kanai et al. (2011), though keeping or breaking a connection is mainly discussed. Kumazawa et al. (2008) also iterates the procedures. Inoue et al. (2008) presents a verification test of a rescheduling support system which predicts the sections and the time span in which the congestion will be heavy.

Schöbel (2001) calls the problem of whether we keep or break a connection of trains and of rescheduling a timetable in case of some known initial delays for trains (and other vehicles in the public transportation) on a network to minimize the total amount of delay over all passengers "delay management." Earlier theoretical work on the delay management is collected in Schöbel (2006, Chapters 6–10). Several integer programming formulations and algorithms are presented there. They include, bicriteria minimization of the total amount of delay and the number of weighted missed connections, which is shown to be $\mathcal{NP}$-hard, as well as bicriteria minimization of the total amount of delay over passengers whose route is in maintained connections and the number of passengers who are involved in missed connections. Gatto et al. (2004) and Gatto et al. (2005) prove that several cases of the delay management are polynomially solvable and that in other cases the decision version is generally $\mathcal{NP}$-complete. The capacity of a track between stations on a network as well as departure orders there are taken into account by Schöbel (2009), and exact and heuristic algorithms are applied to data from the region of Harz, Germany. In most cases, the heuristic algorithms provides near-optimal solutions. Flier et al. (2008) discusses train reordering and change of rolling stock assignment in the delay management and shows its $\mathcal{NP}$-hardness even if we concentrate on only the change of rolling stock assignment. Schachtebeck and Schöbel (2010) also discusses train reordering in the delay management and propose heuristics. Instances from the Harz region in the center of Germany are applied and near-optimal solutions are obtained. Dollevoet et al. (2015) incorporates train reordering, local rerouting and the track capacity inside a station into the original delay management model and applies heuristics to the instances of a part the railway network in the Netherlands. The optimality gap of the solutions is within 0.5 %. Heilporn et al. (2008) presents an alternative formulation which keeps or breaks the connections and minimizes the total amount of delay over all passengers. Instances of a subset of a transportation network around Brussels in Belgium is instantly solved by row generation. De Giovanni et al. (2014) derives new valid inequalities to the formulation presented by Heilporn et al. (2008) and shows that the inequalities define facets of the convex-hull of some special cases. Instances from the medium frequency lines connecting Brussels with some of the most important towns in the Brabant wallon region of Belgium are solved and the optimality gap is halved within 90 seconds. All these models make strong assumptions that an originally planned timetable is cyclic and that after a passenger misses a connection he/she will wait for the same type of train in a next cycle which is supposed to come and go on time.

Dollevoet et al. (2012) concentrates on the connection decisions and rerouting of passengers, provides an integer programming formulation and proves that the problem is generally $\mathcal{NP}$-hard. The complexity of the problem is further studied by Schmidt (2013). Instances of a part the railway network in the Netherlands are solved by an optimization solver within around ten minutes. Dollevoet and Huisman (2014) discusses the model proposed by Dollevoet et al. (2012) and applies heuristics to almost the same instances. Near-optimal solutions are obtained. Dollevoet et al. (2014) iteratively solves a macroscopic delay management model by Dollevoet et al. (2012) on the one hand, and a microscopic train scheduling model by Corman et al. (2011a) on the other hand.

An integrated optimization approach which takes train reordering, local rerouting and passenger rerouting simultaneously into account is proposed by Chigusa et al. (2012). This approach is significant for the delay management since train orders affect passengers' behavior. The rescheduling is formulated as a mixed integer programming problem which minimizes the increased amount of passengers' traveling time. On the other hand, there is a limitation to the passengers' behavior; each passenger is assumed to transfer at most once. Only an imaginary small instance is solved by an optimization solver.

Anderegg et al. (2009) proposes an online optimization version of passengers-oriented rescheduling. At a certain station in a high-frequency bus (or train) system, the delay of the succeeding vehicle is unknown, and the problem is whether the preceding vehicle waits or not so as to minimize the passengers' waiting time. Gatto et al. (2007) presents the "online delay management." These concepts are also discussed by Bauer and Schöbel (2014), Bender et al. (2013), (Annabell) Berger, Blaar, Gebhardt, Müller-Hannemann and Schnee (2011), Kliewer and Suhl (2011) and Krumke et al. (2011). To evaluate and compare different heuristics, (André) Berger, Hoffmann, Lorenz and Stiller (2011) discusses the online delay management and also provides a simulation platform. Caimi et al. (2012) presents a rolling time horizon approach to traffic management in the central railway station area of Berne, Switzerland. This can also be viewed as online optimization. The objective function includes the weighted number of connections to be kept.

We focus on the timetable rescheduling of passenger trains on a line from a macroscopic

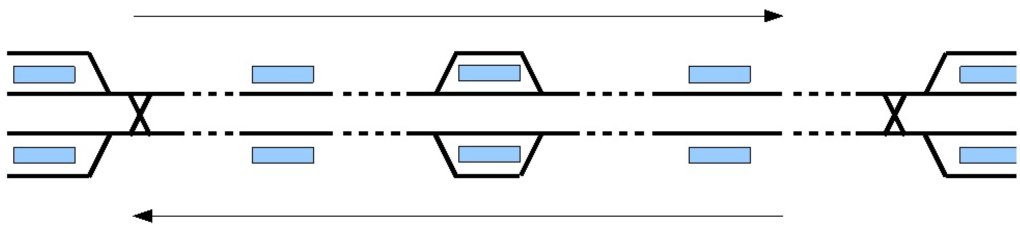

Fig. 3.3 Track layout

viewpoint. We simultaneously handle the passenger rerouting and five detailed rescheduling measures: the reordering, the local rerouting, the retiming, the change of rolling stock assignment and the change of a train type. With all these being incorporated into one mixed integer programming problem, we try to minimize the total increased amount of inconvenience to passengers, which is a valid extension of passengers' arrival delay and is also the dominant part of the terms constituting the disutility value, over all the OD pairs. We solve the offline optimization problem since train services in Japan are known to be punctual on the whole and there is hence a strong possibility that all the trains can be operated in accordance with a rescheduled timetable.

3.3 Problem Description

3.3.1 Railway Line and Timetable

We focus on the timetable rescheduling on railway line whose a track layout is shown in Figure 3.3. This kind of track layout is widely found in urban areas of Japan. Trains run toward one direction on each track on the line. We assume a timetable in which the trains are classified into two types, express and local. The express trains pass through some stations while the local trains stop at every station. The timetable is not limited to be cyclic. The trains can turn back at the left-end and the right-end stations (termini). For each direction towards which the trains run, some of the stations are equipped with only one track. The other stations are equipped with more tracks and the express trains can overtake the local trains there.

3.3.2 Rescheduling Measures

In our timetable rescheduling, we apply a series of detailed rescheduling measures when we observe a certain amount of an initial delay. Its amount is forecast, for instance, based on historical data on the same type of disturbance. One of the forecasts is proposed by Tsuchiya et al. (2006). In our formulations, decision variables on train operations

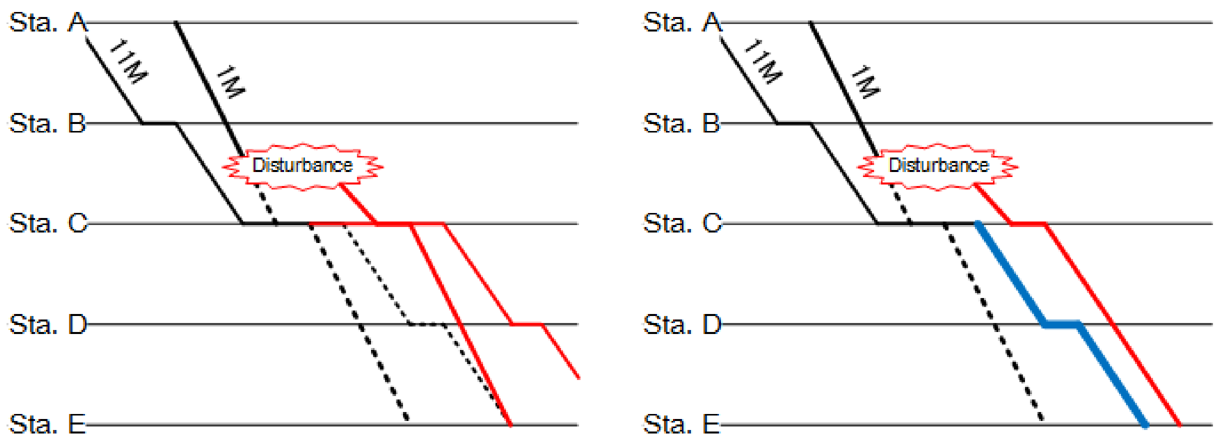

Fig. 3.4 Reordering

corresponds to several detailed rescheduling measures described as follows, which are commonly performed by train dispatchers in Japan and studied in the literature.

Figure 3.4 illustrates an example of train reordering. The dashed lines mean the arrival or departure time of the trains in the current timetable. The left diagram shows that express Train 1 is initially delayed between Stations B and C and that no rescheduling measure is applied. The departure of local Train 11M from Station C will also be delayed consecutively since express Train 1M is supposed to overtake local Train 11M there in the originally planned timetable. The right diagram indicates that the train reordering is performed and local Trains 11M will depart on time.

Another rescheduling measure is local rerouting of a train. We change a route of the train inside stations mainly to avoid a conflict between the train and another one. Since we focus on a track layout displayed in Figure 3.3, local rerouting means change of track assignment which the train passes through or stops at. The left of Figure 3.5 indicates that Trains 22M and 24M stop at the same Track #2 at terminus Stations A and turn back. Since Train 24M can enter the track after Train 22M departs from it, the initial delay of Train 22M will cause that of Train 24M if they are operated according to the current timetable. The right diagram shows that Train 24M will be operated on time if we change the track assignment of Train 24M and let it stop at and turn back from Track #1.

Train retiming is also performed by the dispatchers. One example is to keep, break or make a connection between express and local trains on the same line or trains running on different lines at a station where the two trains stop. In the reordering example shown in Figure 3.4, the connection between express Train 1M and local Train 11M at Station C will be kept if the reordering is not performed. If the connection is broken, passengers who

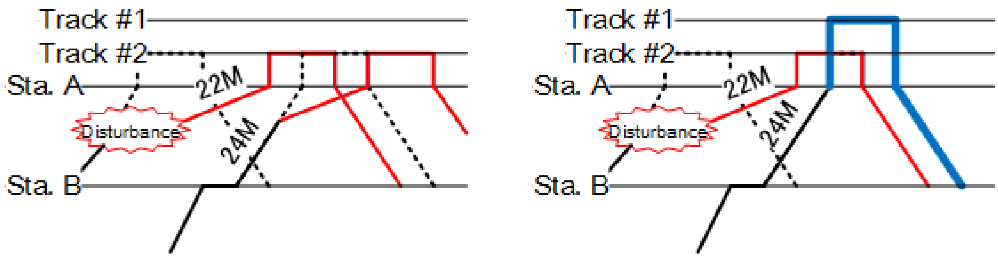

Fig. 3.5 Local rerouting (change of track assignment)

are on express Train 1M and want to go to Station D will be forced to wait for another local train at Station C. Another kind of retiming is performed in a cyclic timetable. Figure 3.6 displays that one train is operated in one cycle time. When Train 33M is initially delayed and no rescheduling measure is applied, the arrival time of Train 35M and Train 37M at Station B will be delayed owing to the track occupation and they are operated so that the delay are recovered. In the right of Figure 3.6, the departure time of Train 33 at Station B will be delayed intentionally until the departure time of Train 35M in the current timetable, and the departure time of Train 35M at Station A will be that of Train 37. In other words, Train 33 will be virtually operated as Train 35M and Train 35M as Train 37M. Then some passengers will do not be feel any delay while others will be involved in further delay. Figure 3.7 shows an example of retiming in a metropolitan area where a larger number of people travel by train. In the left of Figure 3.7, the operation of Train 45M is initially disrupted between Stations A and B. If any rescheduling measure is performed, there will be a long interval between the departure of Trains 43M and 45M at Station C and many passengers will try to board Train 45M one after another at the station, causing congestion and even additional delay of the train. In the right of the Figure 3.7, the departure interval between Train 43M and Train 45M and therefore the congestion of them will be equalized by intentionally delaying the departure of Train 43M at Station C. This type of retiming is performed in a situation where there are not so many passengers since it also contributes to equalize the waiting time of passengers. Concerning these train retiming operations, there is a trade-off between train delay and waiting time of passengers at stations, or that between waiting time of passengers who go from and to different stations and appear at different time. In our rescheduling formulations, we do not explicitly implement these rescheduling measures; it is expected to be achieved if we let the objective of our formulation be passenger-oriented.

Figure 3.8 shows an example of change of a train type. The left of this figure displays the initial departure delay of local Train 51 at Station C, owing to a temporary malfunction of a component of the assigned rolling stock unit for instance, and passengers from Station C

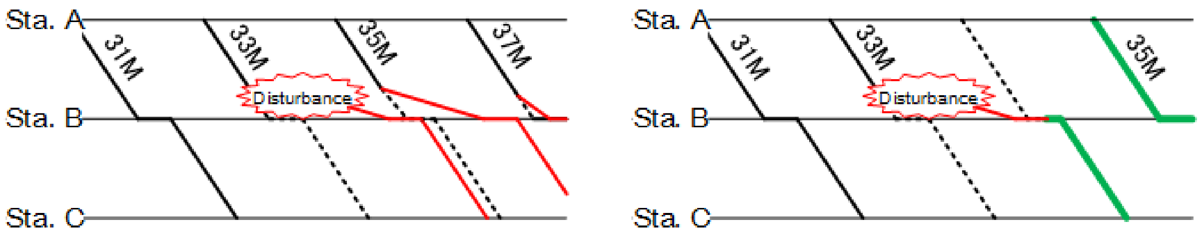

Fig. 3.6 Retiming in cyclic timetable

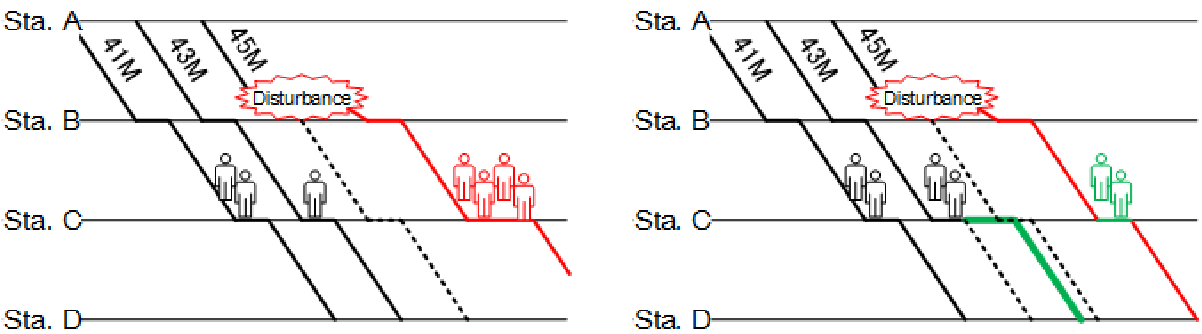

Fig. 3.7 Retiming in metropolitan area

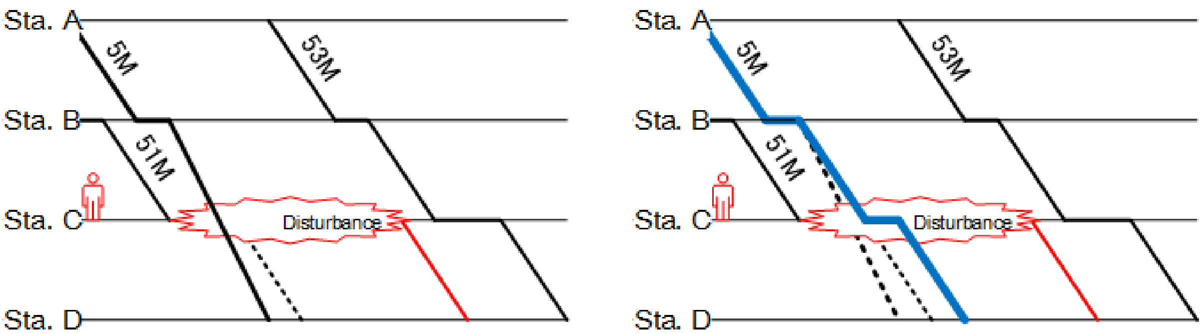

Fig. 3.8 Change of train type

to Station D will be forced to wait for the recovery from the malfunction for a long time. The right of the figure shows that express Train 5M is operated as a local train and stops at Station C. In Japan, the type of trains which have a higher priority is changed into one with a lower priority, and not vice versa. In our rescheduling formulations, we also permit the change of a train type from local to express.

The left of Figure 3.9 displays that the departure of Train 63M at Station A will be delayed since initially delayed Train 62M turns back at the station and then it will be operated as Train 63M. If the rolling stock unit which is being parked at the station and will be operated as Train 65M is assigned to Train 63M and the delayed Train 62M is

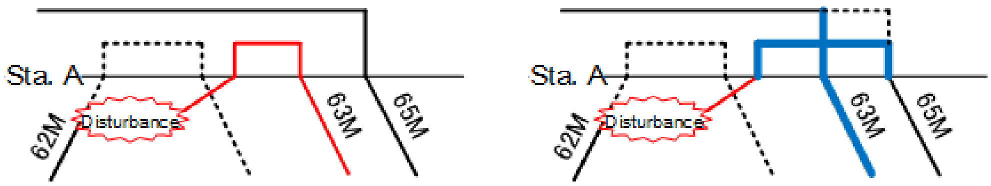

Fig. 3.9 Change of rolling stock assignment

next operated as Train 63M, then the delay will be recovered. Even though the change of rolling stock assignment at a terminus is, strictly speaking, rolling stock rescheduling, it is classified as a timetable rescheduling measure since it prevents the initial delay from spreading. Note that this operation may causes disruption of rolling stock schedule which requires rolling stock rescheduling.

3.3.3 Passenger Behavior

Whether it can be done in real time or not, passenger OD data are electrically available in big cities in Japan since there are automatic ticket gates at every station. When the disruption is small and the delay information has not been announced to (potential) passengers, we can assume that the OD data in the situation are not so different from that on a typical day. When the disruption is large, we have to mix the data of a typical day, the real-time data and a forecast of the future appearance of potential passengers which will be affected by the disruption, based on an approach by Kunimatsu and Hirai (2014) for instance.

We assume that each passenger takes a route which makes inconvenience to him/her be minimized, depending on a rescheduled and announced train operation. Passenger rerouting is possible: he/she may take a different train in a rescheduled timetable from a train which he/she will take in the current timetable and transfer once or more if it leads to the inconvenience. We consider that inconvenience of traveling by train in an undisrupted and rescheduled timetables is given by

$$\begin{aligned}&\text{(traveling time on board) (in minutes)}\\&\quad + \mu \times \text{(waiting time at platforms) (in minutes)} + \nu \times \text{(number of transfers)}. \qquad (3.1)\end{aligned}$$

This expression is the same as the one proposed by Railway Bureau, Ministry of Land, Infrastructure, Transport and Tourism (2012) except that a term on train congestion which he/she experiences is omitted, and is commonly used by civil engineers in Japan to evaluate rail projects. We can omit the term since passengers cannot expect it. If we set $\mu = 1, \nu = 0$, we have the overall traveling time. Railway Bureau, Ministry of Land, Infrastructure, Transport and Tourism (2012) recommends $\mu = 2, \nu = 10$.

3.4 Mixed Integer Programming Formulations

3.4.1 Notation

We formulate the timetable rescheduling as a mixed integer programming problem. Tables 3.1 and 3.2 list the notation. We first introduce sets and their elements commonly used in our mixed integer programming formulations. Let B be a set of train directions (e.g., $B := \{\texttt{Westbound}, \texttt{Eastbound}\}$ in Figure 3.3), and for each direction $b \in B$ its opposite direction $\mathsf{Opp}(b)$ is defined (e.g., $\mathsf{Opp}(\texttt{Westbound}) = \texttt{Eastbound}$).

A set of trains to be rescheduled is denoted by R and $R_b \subseteq R$ is a set of trains traveling in direction b. An ordered pair $R^2_{b\neq}$ means $R^2_{b\neq} := \{(r_1, r_2) \in R^2_b \mid r_1 \neq r_2\}$, i.e., ordered two distinct trains in direction b. Each train $r \in R$ which will arrive at its terminus station may turn back and be operated as another train which has not departed from its starting station at the rescheduling start time. We call it a successor train of r and $R_{\mathsf{Suc}(r)}$ a set of candidate successor trains.

A set of stations is defined as S. Starting and terminus stations of trains traveling in direction $b \in B$ are denoted by $\mathsf{Start}(b)$ and $\mathsf{Term}(b)$, respectively. An element $\mathsf{Next}(b, s)$ of S means a station next to $s \in S$ in direction $b \in B$. Let $S^2_{b<}$ be a set of OD pair in direction b. For passengers who board a train traveling in direction b and whose destination station is $d \in S$, there is an opportunity for a train transfer at some intermediate stations. A set of such stations is denoted by $S_{\mathsf{Tra}(b,d)} \subseteq S$. After a train in direction b leaves station s, there may be a next opportunity for a transfer at a certain station. We name the station $\mathsf{TraNext}(b, s)$. Trains in direction b can pass through or stop at some tracks in station s, and they are denoted by $K^{b,s}$.

Let E be a set of train types (e.g., $E := \{\texttt{Express}, \texttt{Local}\}$). For station $s \in S$, $E^s \subseteq E$ is a set of train types which stop at s. Passengers whose destination station is d and who are now at s (where $(s, d) \in S^2_{b<}$ in direction $b \in B$) will take a train of some types to reach d. The train stops at d or it does not stop but they can alight at d if they transfer to a suitable train at an intermediate station. A set of such train types are defined as $E^{s,d} \subseteq E$. For two train types $e_1, e_2 \in E$ (there is a case with $e_1 = e_2$), we say $e_1 \succsim^E e_2$ if e_1 has a higher priority than or the same priority as e_2. For instance, $\texttt{Express} \succsim^E \texttt{Express}$, $\texttt{Express} \succsim^E \texttt{Local}$ and $\texttt{Local} \succsim^E \texttt{Local}$ hold in general.

A set of discrete time periods is denoted by T. We introduce set F to consider passengers between (o, d_1) and those between (o, d_2) who appear at the same time. When their

Table 3.1 Notation for timetable rescheduling formulations and algorithm (1 of 2)

Sets and Elements	
B	set of train directions
$\mathsf{Opp}(b)$	opposite direction of $b \in B$
R	set of trains to be rescheduled
R_b	set of trains traveling in $b \in B$
$R^2_{b\neq}$	set of ordered pair of distinct trains traveling in $b \in B$
$R_{\mathsf{Suc}(r)}$	set of candidate successor trains of $r \in R$
S	set of stations
$\mathsf{Start}(b)$	starting station of trains traveling in $b \in B$
$\mathsf{Term}(b)$	terminus of trains traveling in $b \in B$
$\mathsf{Next}(b, s)$	station next to $s \in S$ in $b \in B$
$S^2_{b<}$	set of OD pair of stations in $b \in B$
$S_{\mathsf{Tra}(b,\,d)}$	set of stations at which passengers to $d \in S$ in $b \in B$ can transfer
$\mathsf{TraNext}(b, s)$	station next to $s \in S$ in $b \in B$ at which passengers can transfer
$K^{b,s}$	set of tracks at $s \in S$ which trains traveling in $b \in B$ can pass through or stop at
E	set of train types
E^s	set of train types which stop at $s \in S$
$E^{s,d}$	set of train types which passengers can take at s to reach d where $(s, d) \in S^2_{b<}$
$\succsim^E$	priority relation between two train types in E
T	set of discrete time periods
F	set of two pairs $(o, d_1), (o, d_2) \in S^2_{b<}$ with $d_1 \neq d_2$ whose route is same when they appear at same time
Δ^A, Δ^D	set of pairs of trains and stations whose arrival/departure is initially delayed
$\widehat{\Delta}^A, \widehat{\Delta}^D$	set of pairs of trains and stations whose simulated arrival/departure according to current timetable is before rescheduling start time
Parameters	
h_0	rescheduling start time
μ	inconvenience of waiting for one minute at platform
ν	inconvenience of one train transfer
I_{Flex}	flexibility in timetable to be rescheduled

route at o is the same regardless of $d_1 \neq d_2$ (e.g., only local trains stop at o), we let $(o, d_1, d_2) \in F$.

We let h_0 be the rescheduling start time. The parameter μ is the amount of inconvenience of waiting for one minute at a platform and ν is that of one train transfer defined in the expression (3.1). We introduce some flexibility in a timetable to be rescheduled and I_{Flex} indicates the nonnegative parameter for it.

We next introduce several constants. Let A^s_r and D^s_r be arrival and departure times

Table 3.2 Notation for timetable rescheduling formulations and algorithm (2 of 2)

Constants	
A_r^s, D_r^s	arrival/departure time of $r \in R$ at $s \in S$ in originally planned timetable
$A_t^{o,d}$	amount of inconvenience to passengers between $(o,d) \in S_{b<}^2$ appearing at $t \in T$ in originally planned timetable
α_r^s, δ_r^s	arrival/departure time of $(r,s) \in \Delta^A$ / $(r,s) \in \Delta^D$
$\widehat{\alpha}_r^s, \widehat{\delta}_r^s$	simulated arrival/departure time of $(r,s) \in \widehat{\Delta}^A$ / $(r,s) \in \widehat{\Delta}^D$
$\widehat{A}_r^s, \widehat{D}_r^s$	arrival/departure time of $r \in R$ at $s \in S$ in certain rescheduled timetable
$I^{*,*}$	min. interval required between arrivals and departures
M_H, M_I	arbitrary large number concerning headway/inconvenience
$N^{b,s}$	max. number of trains traveling in $b \in B$ allowed to exist between $(s, \mathsf{Next}(b,s))$ on line
$P_t^{o,d}$	number of passengers between $(o,d) \in S_{b<}^2$ appearing at $t \in T$
Decision Variables	
a_r^s, d_r^s	arrival/departure time of $r \in R$ at $s \in S$
$\ell_{r,e}$	1 if type of $r \in R$ is $e \in E$, 0 otherwise
u_r^k	1 if $r \in R_b$ passes through or stops at $k \in K^{b,s}$ at $s \in S$, 0 otherwise
g_{r_1,r_2}	1 if successor of $r_1 \in R$ is $r_2 \in R_{\mathsf{Suc}(r_1)}$, 0 otherwise
x_{r_1,r_2}^s	1 if $r_1 \in R_b$ departs from $s \in S \setminus \{\mathsf{Term}(b)\}$ earlier than $r_2 \in R_b \setminus \{r_1\}$, 0 otherwise
$x_{r_1,r_2}^{\mathsf{Term}(b)}$	1 if $r_1 \in R_b$ arrives at earlier than $r_2 \in R_{\mathsf{Opp}(b)}$ departs from $\mathsf{Term}(b)$, 0 otherwise
$x_{r_2,r_1}^{\mathsf{Term}(b)}$	1 if $r_2 \in R_{\mathsf{Opp}(b)}$ departs from $\mathsf{Term}(b)$ earlier than $r_1 \in R_b$ arrives at, 0 otherwise
n_{r_1,r_2}^s	1 if $N^{b,s} - 1$ trains depart from s after r_1 and before r_2 does, 0 otherwise
$z_{t,r}^{o,d}$	1 if or passengers between $(o,d) \in S_{b<}^2$ appearing at $t \in T$ take $r \in R_b$, 0 otherwise
$z_{r_1,r_2}^{s,d}$	1 if passengers to $d \in S$ at $s \in S_{\mathsf{Tra}(b,\,d)}$ transfer from $r_1 \in R_b$ to $r_2 \in R_b$, 0 otherwise
$\tau_r^{s,d}$	amount of inconvenience to passengers to d when they take $r \in R_b$ at s where $(s,d) \in S_{b<}^2$
$y_{t,r}^{o,d}$	increased amount of inconvenience to passengers between $(o,d) \in S_{b<}^2$ appearing at $t \in T$ taking $r \in R_b$
Formulations	
(DMP)	arrival delay minimization problem
(IMP)	passenger inconvenience minimization problem

of train $r \in R$ at station $s \in S$ in an originally planned timetable. In the timetable, we discuss OD pair $(o,d) \in S_{b<}^2$ in direction $b \in B$ at time $t \in T$, i.e., passengers who appear at station o at time t and travel to station d. We define $A_t^{o,d}$ as the amount of inconvenience which they suffer on their route. Let Δ^A, Δ^D be sets of pairs of trains and stations whose arrival and departure are initially delayed, respectively. If the amount of the initial delays for each Δ^A or Δ^D is given, we can simulate, by applying our rescheduling algorithm or in any manner, the train operations according to the current timetable without rescheduling. We call a timetable in which an arrival or departure time of each train at each station is calculated under these conditions a *simulated timetable*. We let $\widehat{\Delta}^A$ or $\widehat{\Delta}^D$ be sets of pairs of trains and stations whose simulated arrival or departure time is before the rescheduling start time. Pair (r, s) in $\widehat{\Delta}^A$ or $\widehat{\Delta}^D$ means that r is uncontrollable before it departs from or arrives at s. Assume here that a *feasible rescheduled timetable* is given by a certain timetable rescheduling method, e.g., the simulated timetable described above, a train-punctuality-oriented timetable, etc. We let arrival and departure times of train r at station s in such a timetable be $\widehat{A}_r^s, \widehat{D}_r^s$. A Minimum interval required between train arrivals and departures is described as I_*^* depending on a corresponding event. Let M_H, M_I be different arbitrary large numbers concerning headway and inconvenience, respectively. We define $N^{b,s}$ as the maximum number of trains traveling in $b \in B$ which are allowed to exist on the line between s and $\mathsf{Next}(b, s)$. The number of passengers traveling from station o to station d and appearing at the origin station at time t is denoted by $P_t^{o,d}$.

3.4.2 Train Operation Variables and Constraints

We describe variables and integer linear constraints to model feasible train operations. Let a_r^s and d_r^s be continuous variables which indicate arrival and departure time of train $r \in R$ at station $s \in S$, respectively. For the implementation of the rescheduling measures, we introduce binary variables concerning a train type, a train route inside a station, rolling stock assignment at termini, order of trains and a capacity at stations. For each train $r \in R$ and type $e \in E$, we define

$$\ell_{r,e} := \begin{cases} 1 & \text{if the type of } r \text{ is } e, \\ 0 & \text{otherwise.} \end{cases}$$

For each direction $b \in B$, train $r \in R_b$ in the direction, station $s \in S$ and track $k \in K^{b,s}$ in the station which the train can pass through or stop at, we set

$$u_r^k := \begin{cases} 1 & \text{if } r \text{ passes through or stops at } k, \\ 0 & \text{otherwise.} \end{cases}$$

For each train $r_1 \in R$ and its candidate successor train $r_2 \in R_{\mathsf{Suc}(r_1)}$, we let

$$g_{r_1,r_2} := \begin{cases} 1 & \text{if } r_2 \text{ is a successor of } r_1, \\ 0 & \text{otherwise.} \end{cases}$$

For each direction $b \in B$, ordered pair of distinct trains $(r_1, r_2) \in R^2_{b\neq}$ in the direction and station $s \in S \setminus \{\mathsf{Term}(b)\}$ other than the terminus, we define

$$x^s_{r_1,r_2} := \begin{cases} 1 & \text{if } r_1 \text{ departs from } s \text{ earlier than } r_2 \text{ does,} \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, for each direction $b \in B$, train $r_1 \in R_b$ in the direction, its terminus $\mathsf{Term}(b)$ and train $r_2 \in R_{\mathsf{Opp}(b)}$ in the opposite direction,

$$x^{\mathsf{Term}(b)}_{r_1,r_2} := \begin{cases} 1 & \text{if } r_1 \text{ arrives at } s \text{ earlier than } r_2 \text{ departs from,} \\ 0 & \text{otherwise,} \end{cases}$$

$$x^{\mathsf{Term}(b)}_{r_2,r_1} := \begin{cases} 1 & \text{if } r_2 \text{ departs from } s \text{ earlier than } r_1 \text{ arrives at,} \\ 0 & \text{otherwise.} \end{cases}$$

For each direction $b \in B$, ordered pair of distinct trains $(r_1, r_2) \in R^2_{b\neq}$ in the direction and station $s \in S \setminus \{\mathsf{Term}(b)\}$ other than the terminus, we let

$$n^s_{r_1,r_2} := \begin{cases} 1 & \text{if } N^{b,s} - 1 \text{ or more trains depart from } s \text{ after } r_1 \text{ and before } r_2 \text{ does,} \\ 0 & \text{otherwise.} \end{cases}$$

We then give train operation constraints for each single train. One among the train type set members is selected:

$$\sum_{e \in E} \ell_{r,e} = 1 \qquad \forall r \in R. \tag{3.2}$$

A train passes through or stops at one track in a station:

$$\sum_{k \in K^{b,s}} u^k_r = 1 \qquad \forall b \in B \ \ \forall r \in R_b \ \ \forall s \in S. \tag{3.3}$$

Each train may turn back and be operated as another one train. When a train is a candidate successor of several trains, exactly one predecessor is determined. They are modeled by

$$\sum_{r_2 \in R_{\mathsf{Suc}(r_1)}} g_{r_1,r_2} \leq 1 \qquad \forall r_1 \in R, \tag{3.4}$$

$$\sum_{r_1 \in \{r \in R \mid r_2 \in R_{\mathsf{Suc}(r)}\}} g_{r_1,r_2} = 1 \qquad \forall r_2 \in R \text{ such that } \{r \in R \mid r_2 \in R_{\mathsf{Suc}(r)}\} \neq \varnothing. \tag{3.5}$$

If $g_{r_1,r_2} = 1$ holds, i.e., the successor of train r_1 is r_2, then they occupy the same track at the terminus (otherwise there is no such restriction):

$$-(1 - g_{r_1,r_2}) \leq u^k_{r_1} - u^k_{r_2} \leq 1 - g_{r_1,r_2}$$
$$\forall b \in B \;\; \forall r_1 \in R_b \;\; \forall r_2 \in R_{\mathsf{Suc}(r_1)} \;\; \forall k \in K^{b,\mathsf{Term}(b)}. \tag{3.6}$$

The trains do not arrive at or depart from the stations earlier than they do in the originally planned timetable. Additionally, we impose the constraint that they do not arrive or depart later than specified time either, on account of intractability of large mixed integer programming problems in terms of solution space. There is a trade-off between flexibility in a timetable to be rescheduled and the tractability of the problem. We let the time be the sum of the arrival or departure time in a certain feasible rescheduled timetable and the flexibility parameter I_{Flex}:

$$A^s_r \leq a^s_r \leq \widehat{A}^s_r + I_{\mathsf{Flex}} \qquad \forall b \in B \;\; \forall r \in R_b \;\; \forall s \in S \setminus \{\mathsf{Start}(b)\}, \tag{3.7}$$

$$D^s_r \leq d^s_r \leq \widehat{D}^s_r + I_{\mathsf{Flex}} \qquad \forall b \in B \;\; \forall r \in R_b \;\; \forall s \in S \setminus \{\mathsf{Term}(b)\}. \tag{3.8}$$

For $(r, s) \in \Delta^A$ or $(r, s) \in \Delta^D$, let α^s_r, δ^s_r be arrival or departure time of initially delayed train r at station s. Then we consider the initial delay in our formulation as follows:

$$a^s_r \geq \alpha^s_r \qquad \forall (r, s) \in \Delta^A, \tag{3.9}$$

$$d^s_r \geq \delta^s_r \qquad \forall (r, s) \in \Delta^D. \tag{3.10}$$

Similarly for $(r, s) \in \widehat{\Delta}^A$ or $(r, s) \in \widehat{\Delta}^D$, let $\widehat{\alpha}^s_r, \widehat{\delta}^s_r$ be arrival or departure time in a simulated timetable. Since (r, s) is an arrival or departure event before the rescheduling start time, we assume that its time is the simulated one:

$$a^s_r = \widehat{\alpha}^s_r \qquad \forall (r, s) \in \widehat{\Delta}^A, \tag{3.11}$$

$$d^s_r = \widehat{\delta}^s_r \qquad \forall (r, s) \in \widehat{\Delta}^D. \tag{3.12}$$

Minimum running time $I^{b,s}_e$ of train r traveling in direction $b \in B$ between two stations

$(s, \mathsf{Next}(b, s))$ depends on its train type e. The constraints can be aggregated from the constraint (3.2), and is modeled by

$$a_r^{\mathsf{Next}(b,\,s)} - d_r^s \geq \sum_{e \in E} I_e^{b,s} \ell_{r,e} \qquad \forall b \in B \;\; \forall r \in R_b \;\; \forall s \in S \setminus \{\mathsf{Term}(b)\}. \tag{3.13}$$

Minimum dwell time I^s at station s is required if a train stops there. If it passes, then the arrival and departure times are equivalent. We introduce the large constant M_H and linearize the constraints, according to the technique presented by Williams (2013) for instance:

$$\sum_{e \in E^s} I^s \ell_{r,e} \leq d_r^s - a_r^s \leq M_H (\sum_{e \in E^s} \ell_{r,e}) \qquad \forall r \in R \;\; \forall s \in S. \tag{3.14}$$

Turn back time $I_{\mathsf{Suc}}^{\mathsf{Term}(b)}$ from r_1 to r_2 at terminus $\mathsf{Term}(b)$ is given by

$$d_{r_2}^{\mathsf{Term}(b)} - a_{r_1}^{\mathsf{Term}(b)} \geq I_{\mathsf{Suc}}^{\mathsf{Term}(b)} - M_H(1 - g_{r_1,r_2}) \quad \forall b \in B \;\; \forall r_1 \in R_b \;\; \forall r_2 \in R_{\mathsf{Suc}(r_1)}. \tag{3.15}$$

We let arrival and departure times of a train at its terminus be equivalent to those of its successor train:

$$\begin{aligned} -M_H(1 - g_{r_1,r_2}) \leq a_{r_2}^{\mathsf{Term}(b)} - a_{r_1}^{\mathsf{Term}(b)} &\leq M_H(1 - g_{r_1,r_2}) \\ \forall b \in B \;\; \forall r_1 \in R_b \;\; &\forall r_2 \in R_{\mathsf{Suc}(r_1)}, \end{aligned} \tag{3.16}$$

$$\begin{aligned} -M_H(1 - g_{r_1,r_2}) \leq d_{r_2}^{\mathsf{Term}(b)} - d_{r_1}^{\mathsf{Term}(b)} &\leq M_H(1 - g_{r_1,r_2}) \\ \forall b \in B \;\; \forall r_1 \in R_b \;\; &\forall r_2 \in R_{\mathsf{Suc}(r_1)}. \end{aligned} \tag{3.17}$$

We present operation constraints between two trains. Exactly one of two trains r_1, r_2 in direction b departs earlier at the stations other than terminus $\mathsf{Term}(b)$. At the terminus, a train arrives earlier than another train departs or vice versa:

$$x_{r_1,r_2}^s + x_{r_2,r_1}^s = 1 \qquad \forall b \in B \;\; \forall (r_1, r_2) \in R_{b\neq}^2 \;\; \forall s \in S \setminus \{\mathsf{Term}(b)\}, \tag{3.18}$$

$$x_{r_1,r_2}^{\mathsf{Term}(b)} + x_{r_2,r_1}^{\mathsf{Term}(b)} = 1 \qquad \forall b \in B \;\; \forall r_1 \in R_b \;\; \forall r_2 \in R_{\mathsf{Opp}(b)}. \tag{3.19}$$

Assume here that the priority of type e_1 of r_1 is higher than or the same as that of e_2 of r_2. If r_1 departs earlier than r_2 from station s, then their departing order is maintained at the next stations:

$$x_{r_1,r_2}^{\mathsf{Next}(b,\,s)} \geq x_{r_1,r_2}^{s} - (2 - \ell_{r_1,e_1} - \ell_{r_2,e_2})$$
$$\forall b \in B \ \ \forall (r_1,r_2) \in R_{b\neq}^2 \ \ \forall s \in S \setminus \{\mathsf{Term}(b)\} \text{ such that } \mathsf{Next}(b,s) \neq \mathsf{Term}(b)$$
$$\forall (e_1,e_2) \in E^2 \text{ such that } e_1 \succsim^{E} e_2. \quad (3.20)$$

If r_2 is a successor of r_1, then r_1 arrives earlier than r_2 departs:

$$g_{r_1,r_2} \leq x_{r_1,r_2}^{\mathsf{Term}(b)} \qquad \forall b \in B \ \ \forall r_1 \in R_b \ \ \forall r_2 \in R_{\mathsf{Suc}(r_1)}. \quad (3.21)$$

For any two departures of trains r_1, r_2 in direction b from station s, the number of trains which succeed r_2 subtracting from the number of trains which succeed r_1 is equal to one added to the number of other trains which depart after r_1 and before r_2 does. If it is $N^{b,s}$ or more, then we make the corresponding binary variable n_{r_1,r_2}^{s} be one by the following inequality:

$$n_{r_1,r_2}^{s} \geq \Big(\sum_{r \in R_b \setminus \{r_1\}} x_{r_1,r}^{s} - \sum_{r \in R_b \setminus \{r_2\}} x_{r_2,r}^{s} - N^{b,s} + 1\Big)/(|R_b| - 1)$$
$$\forall b \in B \ \ \forall (r_1,r_2) \in R_{b\neq}^2 \ \ \forall s \in S \setminus \{\mathsf{Term}(b)\}. \quad (3.22)$$

Minimum headway between arrivals of trains ($I_{\mathsf{AA}}^{\mathsf{Next}(b,\,s)}$) and departures of them (I_{DD}^{s}) are modeled by:

$$a_{r_2}^{\mathsf{Next}(b,\,s)} - a_{r_1}^{\mathsf{Next}(b,\,s)} \geq I_{\mathsf{AA}}^{\mathsf{Next}(b,\,s)} - M_H(1 - x_{r_1,r_2}^{s})$$
$$\forall b \in B \ \ \forall (r_1,r_2) \in R_{b\neq}^2 \ \ \forall s \in S \setminus \{\mathsf{Term}(b)\}, \quad (3.23)$$
$$d_{r_2}^{s} - d_{r_1}^{s} \geq I_{\mathsf{DD}}^{s} - M_H(1 - x_{r_1,r_2}^{s})$$
$$\forall b \in B \ \ \forall (r_1,r_2) \in R_{b\neq}^2 \ \ \forall s \in S \setminus \{\mathsf{Term}(b)\}. \quad (3.24)$$

If arriving and departing trains try to occupy the same track at station $\bar{s}$, then minimum headway $I_{\mathsf{AD}}^{\bar{s}}$ is required. Two forms of constraints are necessary corresponding to the cases where $\bar{s}$ is a starting station or not:

$$a_{r_2}^{\mathsf{Start}(b)} - d_{r_1}^{\mathsf{Start}(b)} \geq I_{\mathsf{AD}}^{\mathsf{Start}(b)} - M_H(3 - x_{r_1,r_2}^{\mathsf{Start}(b)} - u_{r_1}^{k} - u_{r_2}^{k})$$
$$\forall b \in B \ \ \forall (r_1,r_2) \in R_{b\neq}^2 \ \ \forall k \in K^{b,\mathsf{Start}(b)}, \quad (3.25)$$
$$a_{r_2}^{\mathsf{Next}(b,\,s)} - d_{r_1}^{\mathsf{Next}(b,\,s)} \geq I_{\mathsf{AD}}^{\mathsf{Next}(b,\,s)} - M_H(3 - x_{r_1,r_2}^{s} - u_{r_1}^{k} - u_{r_2}^{k})$$
$$\forall b \in B \ \ \forall (r_1,r_2) \in R_{b\neq}^2 \ \ \forall s \in S \setminus \{\mathsf{Term}(b)\} \ \ \forall k \in K^{b,\mathsf{Next}(b,\,s)}. \quad (3.26)$$

At the crossing near terminus $\mathsf{Term}(b)$, minimum headway $I_{\mathsf{AD}}^{\mathsf{Term}(b)}$ or $I_{\mathsf{DA}}^{\mathsf{Term}(b)}$ between two trains traveling in different directions is required:

$$a_{r_1}^{\mathsf{Term}(b)} - d_{r_2}^{\mathsf{Term}(b)} \geq I_{\mathsf{AD}}^{\mathsf{Term}(b)} - M_H(1 - x_{r_2,r_1}^{\mathsf{Term}(b)})$$
$$\forall b \in B \ \ \forall r_1 \in R_b \ \ \forall r_2 \in R_{\mathsf{Opp}(b)}, \tag{3.27}$$
$$d_{r_2}^{\mathsf{Term}(b)} - a_{r_1}^{\mathsf{Term}(b)} \geq I_{\mathsf{DA}}^{\mathsf{Term}(b)} - M_H(1 - x_{r_1,r_2}^{\mathsf{Term}(b)})$$
$$\forall b \in B \ \ \forall r_1 \in R_b \ \ \forall r_2 \in R_{\mathsf{Opp}(b)}. \tag{3.28}$$

If there are $N^{b,s}$ trains on the line between $(s, \mathsf{Next}(b, s))$ whose front is r_1, which is ensured by $n_{r_1,r_2}^s = 1$, then r_2 can leave s after r_1 arrives at $\mathsf{Next}(b, s)$ and minimum headway $I_{\mathsf{DA}}^{b,s}$ elapses:

$$d_{r_2}^s - a_{r_1}^{\mathsf{Next}(b,\,s)} \geq I_{\mathsf{DA}}^{b,s} - M_H(1 - n_{r_1,r_2}^s)$$
$$\forall b \in B \ \ \forall (r_1, r_2) \in R_{b\neq}^2 \ \ \forall s \in S \setminus \{\mathsf{Term}(b)\}. \tag{3.29}$$

3.4.3 Passenger Behavior Variables and Constraints

We introduce two kinds of 0-1 variables to model passenger's behavior, and two continuous variable concerning inconvenience to them. For each direction $b \in B$, OD pair of stations $(o, d) \in S_{b<}^2$, time $t \in T$ and train $r \in R_b$, we define

$$z_{t,r}^{o,d} := \begin{cases} 1 & \text{if passengers from } o \text{ to } d \text{ appearing at } t \text{ take } r, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, for each direction $b \in B$, destination $d \in S$, intermediate station $s \in S_{\mathsf{Tra}(b,\,d)}$ and pair of trains $(r_1, r_2) \in R_b^2$ in the direction,

$$z_{r_1,r_2}^{s,d} := \begin{cases} 1 & \text{if passengers to } d \text{ transfer from } r_1 \text{ to } r_2 \text{ at } s, \\ 0 & \text{otherwise.} \end{cases}$$

Note that there is a case with $r_1 = r_2$, i.e., they do not transfer. For each direction $b \in B$, OD pair of stations $(s, d) \in S_{b<}^2$ and train $r \in R_b$, let $\tau_r^{s,d}$ be the amount of inconvenience to the passengers who take r at station s to go to their destination d without waiting for the train at the platform. We also introduce a variable which indicates the increased amount of inconvenience which the passengers suffer in an optimal rescheduled timetable, i.e., the positive difference between the amount of inconvenience in the rescheduled timetable and that in the planned timetable. Note that we regard the value as zero if the former is smaller than the latter. For each direction $b \in B$, OD pair $(o, d) \in S^2_{b<}$, time $t \in T$ and train $r \in R_b$, we denote by $y^{o,d}_{t,r}$ the increased amount of inconvenience to passengers who appear at station o and at time t, go to their destination d and take train r at o.

We then describe passenger behavior constraints at his/her origin station. Firstly, only one train is selected:

$$\sum_{r \in R_b} z^{o,d}_{t,r} = 1 \qquad \forall b \in B \;\; \forall (o, d) \in S^2_{b<} \;\; \forall t \in T. \tag{3.30}$$

The following inequality indicates $d^o_r < t \Rightarrow z^{o,d}_{t,r} = 0$, i.e., the passengers cannot board any train which departs from the origin station before they appear there:

$$z^{o,d}_{t,r} \leq d^o_r / t \qquad \forall b \in B \;\; \forall (o, d) \in S^2_{b<} \;\; \forall t \in T \;\; \forall r \in R_b. \tag{3.31}$$

If a train passes through their origin station, or they cannot alight at their destination station by taking the train and even transferring to any other train at an intermediate station, then the value $z^{o,d}_{t,r}$ must be zero:

$$z^{o,d}_{t,r} \leq 1 - \ell_{r,e} \qquad \forall b \in B \;\; \forall (o, d) \in S^2_{b<} \;\; \forall t \in T \;\; \forall r \in R_b \;\; \forall e \in E \setminus E^{o,d}. \tag{3.32}$$

At the intermediate stations, there are transfer constraints which are similar to the constraints (3.30)–(3.32), where I_{Tra} is time required for the passengers to transfer:

$$\begin{aligned}
&\sum_{r_2 \in R_b} z^{s,d}_{r_1,r_2} = 1 \\
&\qquad\qquad \forall b \in B \;\; \forall d \in S \;\; \forall s \in S_{\mathsf{Tra}(b,\,d)} \;\; \forall r_1 \in R_b, && (3.33)\\
&d^s_{r_2} - a^s_{r_1} \geq I_{\mathsf{Tra}} - M_H(1 - z^{s,d}_{r_1,r_2}) \\
&\qquad\qquad \forall b \in B \;\; \forall d \in S \;\; \forall s \in S_{\mathsf{Tra}(b,\,d)} \;\; \forall (r_1, r_2) \in R^2_{b\neq}, && (3.34)\\
&z^{s,d}_{r_1,r_2} \leq 1 - \ell_{r_2,e} \\
&\qquad\qquad \forall b \in B \;\; \forall d \in S \;\; \forall s \in S_{\mathsf{Tra}(b,\,d)} \;\; \forall (r_1, r_2) \in R^2_b \;\; \forall e \in E \setminus E^{s,d}. && (3.35)
\end{aligned}$$

The amount of inconvenience $\tau^{s,d}_r$ is uniquely determined when there is no opportunity for a transfer on the way from station s to station d. It is simply the traveling time on board:

$$\tau_r^{s,d} = a_r^d - d_r^s$$
$$\forall b \in B \;\; \forall (s,d) \in S_{b<}^2 \text{ such that } \mathsf{TraNext}(b,s) \notin S_{\mathsf{Tra}(b,\,d)} \;\; \forall r \in R_b. \quad (3.36)$$

At intermediate station $\mathsf{TraNext}(b,s)$ where a train transfer is possible on the way from s to d, the amount of inconvenience to the passengers may or may not be decreased by changing trains. Consider a case where the passengers are still on the same train r. Then, $\tau_r^{s,d}$ consists of, the time they spend on board between the departure from s and the departure from $\mathsf{TraNext}(b,s)$, and the amount of inconvenience $\tau_r^{\mathsf{TraNext}(b,\,s),d}$ from $\mathsf{TraNext}(b,s)$ to d. We model the constraint by introducing the large constant M_I:

$$\tau_r^{s,d} \geq a_r^{\mathsf{TraNext}(b,\,s)} - d_r^s$$
$$+ \left(d_r^{\mathsf{TraNext}(b,\,s)} - a_r^{\mathsf{TraNext}(b,\,s)}\right) + \tau_r^{\mathsf{TraNext}(b,\,s),d} - M_I\left(1 - z_{r,r}^{\mathsf{TraNext}(b,\,s),d}\right)$$
$$\forall b \in B \;\; \forall (s,d) \in S_{b<}^2 \text{ such that } \mathsf{TraNext}(b,s) \in S_{\mathsf{Tra}(b,\,d)} \;\; \forall r \in R_b. \quad (3.37)$$

Suppose that the passengers transfer from r_1 to r_2. Then the waiting time multiplied by μ instead of the dwell time and the train transfer penalty ν are imposed:

$$\tau_{r_1}^{s,d} \geq a_{r_1}^{\mathsf{TraNext}(b,\,s)} - d_{r_1}^s$$
$$+ \mu \times \left(d_{r_2}^{\mathsf{TraNext}(b,\,s)} - a_{r_1}^{\mathsf{TraNext}(b,\,s)}\right) + \nu + \tau_{r_2}^{\mathsf{TraNext}(b,\,s),d} - M_I\left(1 - z_{r_1,r_2}^{\mathsf{TraNext}(b,\,s),d}\right)$$
$$\forall b \in B \;\; \forall (s,d) \in S_{b<}^2 \text{ such that } \mathsf{TraNext}(b,s) \in S_{\mathsf{Tra}(b,\,d)} \;\; \forall (r_1,r_2) \in R_{b\neq}^2. \quad (3.38)$$

Finally we calculate the increased amount of inconvenience to passengers $y_{t,r}^{o,d}$. It is given by $y_{t,r}^{o,d} = z_{t,r}^{o,d} \times \max\{\mu \times (d_r^o - t) + \tau_r^{o,d} - A_t^{o,d}, 0\}$. We first consider a case where $z_{t,r}^{o,d} = 1$, i.e., the passengers who appear at the origin station o at the time t take the train r to go to their destination d. The term $\mu \times (d_r^o - t)$ is the inconvenience of waiting for r and $\tau_r^{o,d}$ is the inconvenience which they suffer after taking r. We subtract the amount of inconvenience in the planned timetable from the sum of these two terms, which means the total amount of inconvenience in the disrupted situation. When its value is negative, we regard the "increased" amount of inconvenience as zero. We next consider the case where $z_{t,r}^{o,d} = 0$, which indicates that they do not take r at their origin station. Then the inconvenience by taking this train is zero. This complicated equation is linearized in the following way:

$$y^{o,d}_{t,r} \geq \mu \times (d^o_r - t) + \tau^{o,d}_r - A^{o,d}_t - M_I(1 - z^{o,d}_{t,r})$$
$$\forall b \in B \ \ \forall (o,d) \in S^2_{b<} \ \ \forall t \in T \ \ \forall r \in R_b, \tag{3.39}$$
$$y^{o,d}_{t,r} \geq 0$$
$$\forall b \in B \ \ \forall (o,d) \in S^2_{b<} \ \ \forall t \in T \ \ \forall r \in R_b. \tag{3.40}$$

3.4.4 Additional Constraints

We add several constraints to obtain an optimal solution to in shorter time. Although these constraints are redundant, i.e., the feasible region of the mixed integer programming problem are equivalent with or without them, those of the linear programming relaxation problems may be different. The constraints presented in this subsection are expected to give a tighter lower bound in the branch-and-cut process in general mixed integer programming algorithms.

Let the type of two trains r_1, r_2 be the same and $t \leq D^o_{r_1}$ holds, i.e., passengers appearing at t do not miss r_1 from $D^o_{r_1} \leq d^o_{r_1}$ in the constraint (3.8). If $x^o_{r_1,r_2} = 1$ then r_1 also departs earlier than r_2 from the stations afterwards from the constraint (3.20). Therefore they do not take r_2:

$$z^{o,d}_{t,r_2} \leq 3 - x^o_{r_1,r_2} - \ell_{r_1,e} - \ell_{r_2,e}$$
$$\forall b \in B \ \ \forall (o,d) \in S^2_{b<} \ \ \forall t \in T \ \ \forall (r_1,r_2) \in R^2_{b\neq} \text{ such that } t \leq D^o_{r_1} \ \ \forall e \in E^o. \tag{3.41}$$

We next consider a case where the passengers can travel between (o,d) by boarding single train r_1, r_2 stops at o and r_1 arrives at d earlier than r_2. When $\mu = 1$, i.e., the inconvenience they suffer on their route consists of their traveling time and a transfer, taking r_1 is a better choice; the traveling time by taking r_1 is shorter than that by taking r_2 and r_1 stops at o and d. Hence, they do not take r_2 at o. This situation is modeled as follows:

$$z^{o,d}_{t,r_2} \leq 3 - x^{\bar{s}}_{r_1,r_2} - \ell_{r_1,e_1} - \ell_{r_2,e_2}$$
$$\forall b \in B \ \ \forall (o,d) \in S^2_{b<} \ \ \bar{s} \in S \text{ such that } \mathsf{Next}(b,\bar{s}) = d \ \ \forall t \in T$$
$$\forall (r_1,r_2) \in R^2_{b\neq} \text{ such that } t \leq D^o_{r_1} \ \ \forall e_1 \in E^o \cap E^d \ \ e_2 \in E^o \text{ such that } e_1 \neq e_2. \tag{3.42}$$

The passengers do not transfer to a train of the same type:

$$z^{s,d}_{r_1,r_2} \leq 2 - \ell_{r_1,e} - \ell_{r_2,e} \quad \forall b \in B \ \ \forall d \in S \ \ \forall s \in S_{\mathsf{Tra}(b,\, d)} \ \ \forall (r_1, r_2) \in R^2_{b\neq} \ \ \forall e \in E. \quad (3.43)$$

If the passengers transfer from r_1 to another train, then it must be done at only one station:

$$\sum_{s \in S_{\mathsf{Tra}(b,\, d)}} \sum_{r_2 \in R_b \setminus \{r_1\}} z^{s,d}_{r_1,r_2} \leq 1 \qquad \forall b \in B \ \ \forall d \in S \ \ \forall r_1 \in R_b. \quad (3.44)$$

Let r_1, r_2 stop at d and r_1 arrives at d earlier. Then the sum of the waiting time of r_2 after alighting from r_1 for the transfer and the traveling time on board by r_2 is longer than the traveling time on board by r_1. When $\mu \geq 1$, any passenger do not transfer to r_2 since it brings more inconvenience. This is expressed as follows:

$$\begin{aligned} &z^{s,d}_{r_1,r_2} \leq 3 - x^{\bar{s}}_{r_1,r_2} - \ell_{r_1,e_1} - \ell_{r_2,e_2} \\ &\qquad \forall b \in B \ \ \forall d \in S \ \ \forall s \in S_{\mathsf{Tra}(b,\, d)} \ \ \bar{s} \in S \text{ such that } \mathsf{Next}(b, \bar{s}) = d \\ &\qquad \forall (r_1, r_2) \in R^2_{b\neq} \ \ \forall e_1, e_2 \in E^d. \end{aligned} \quad (3.45)$$

Consider passengers between (o, d_1) and those between (o, d_2) who appear at the same time. When $(o, d_1, d_2) \in F$ holds, i.e., their route at o is the same, two such OD pairs can be aggregated as follows:

$$\begin{aligned} &z^{o,d_1}_{t,r} = z^{o,d_2}_{t,r} \\ &\qquad \forall b \in B \ \ \forall (o, d_1), (o, d_2) \in S^2_{b<} \text{ such that } d_1 \neq d_2 \text{ and } (o, d_1, d_2) \in F \\ &\qquad \forall t \in T \ \ \forall r \in R_b. \end{aligned} \quad (3.46)$$

Similarly at the intermediate stations, we have

$$\begin{aligned} &z^{s,d_1}_{r_1,r_2} = z^{s,d_2}_{r_1,r_2} \\ &\qquad \forall b \in B \ \ \forall (s, d_1), (s, d_2) \in S^2_{b<} \text{ such that } d_1 \neq d_2 \text{ and } (s, d_1, d_2) \in F \\ &\qquad \forall (r_1, r_2) \in R^2_b. \end{aligned} \quad (3.47)$$

3.4.5 Objective Functions

We consider two objective functions in our timetable rescheduling; the one simply minimizes the total amount of arrival delays and ignores the passenger behavior model. We call the mixed integer programming problem the arrival delay minimization problem (DMP) and the formulation is as follows:

$$(DMP) \quad \left| \begin{array}{ll} \text{minimize} & \displaystyle\sum_{b \in B} \sum_{r \in R_b} \sum_{s \in S \setminus \{\mathsf{Start(b)}\}} (a_r^s - A_r^s) \\ \text{subject to} & (3.2)\text{–}(3.29), \\ & \ell_{r,e}, u_r^{s,k}, g_{r_1,r_2}, x_{r_1,r_2}^s, n_{r_1,r_2}^s \in \{0,1\}. \end{array} \right.$$

Time is a unit of the objective value of (DMP) and its lower bound is zero from the constraint (3.7).

The other is to minimize the sum of increased amount of inconvenience over all the passengers and it is what we think is of crucial importance. We name it the passenger inconvenience minimization problem (IMP) and present the formulation below:

$$(IMP) \quad \left| \begin{array}{ll} \text{minimize} & \displaystyle\sum_{b \in B} \sum_{(o,d) \in S_{b<}^2} \sum_{t \in T} P_t^{o,d} (\sum_{r \in R_b} y_{t,r}^{o,d}) \\ \text{subject to} & (3.2)\text{–}(3.47), \\ & \ell_{r,e}, u_r^{s,k}, g_{r_1,r_2}, x_{r_1,r_2}^s, n_{r_1,r_2}^s, z_{t,r}^{o,d}, z_{r_1,r_2}^{s,d} \in \{0,1\}. \end{array} \right.$$

The increased amount of inconvenience times the number of passengers is a unit of the objective value of (IMP) and its lower bound of is also zero from the constraint (3.40).

3.5 Algorithm

3.5.1 Overall Algorithm and Passenger Inconvenience Calculation in Originally Planned Timetable

We set all the necessary input data and minimize further inconvenience to passengers caused by disturbance. The overall algorithm is displayed in Figure 3.10.

At Step 1, we calculate the amount of inconvenience to passengers in the originally planned timetable before there occurs a delay, since our objective is to minimize the positive difference between the inconvenience in a rescheduled timetable and that in the

Step 1: *Passenger Inconvenience Calculation in Originally Planned Timetable.*
Construct time-space graph from originally planned timetable and T.
Set amount of inconvenience on each edge.
For each $b \in B, (o, d) \in S_{b<}^2, t \in T$,
Find shortest path of passengers from o to d appearing at t.
Let shortest path length be $A_t^{o,d}$.
Step 2: *Delay Spread Simulation.*
For each $(r^A, s^A) \in \Delta^A, (r^D, s^D) \in \Delta^D$, input $\alpha_{r^A}^{s^A}, \delta_{r^D}^{s^D}$.
For each $r \in R, s \in S$, set $\widehat{A}_r^s, \widehat{D}_r^s := \infty$.
Set $\widehat{\Delta}^A = \varnothing, \widehat{\Delta}^D = \varnothing$.
Calculate M_H.
Solve (DMP) with $\ell_{r,e}, u_r^{s,k}, g_{r_1,r_2}, x_{r_1,r_2}^s$ fixed.
Step 3: *Solving Arrival Delay Minimization Problem.*
Let $\check{a}_r^s, \check{d}_r^s$ be solution to (DMP) obtained at Step 2.
Set h_0.
Set $\widehat{\Delta}^A := \{(r, s) \in R \times S \mid \check{a}_r^s < h_0\}, \widehat{\Delta}^D := \{(r, s) \in R \times S \mid \check{d}_r^s < h_0\}$.
For each $(r^A, s^A) \in \widehat{\Delta}^A, (r^D, s^D) \in \widehat{\Delta}^D$, set $\widehat{\alpha}_{r^A}^{s^A} := \check{a}_r^s, \widehat{\delta}_{r^D}^{s^D} := \check{d}_r^s$.
Solve (DMP).
Step 4: *Solving Passenger Inconvenience Minimization Problem.*
Let $\check{a}_r^s, \check{d}_r^s$ be solution to (DMP) obtained at Step 3.
For each $r \in R, s \in S$, set $\widehat{A}_r^s := \check{a}_r^s, \widehat{D}_r^s := \check{d}_r^s$.
Give I_{Flex}.
Update M_H.
Calculate M_I.
Solve (IMP).

Fig. 3.10 Timetable rescheduling algorithm

planned timetable. Namely, we compute $A_t^{o,d}$ for each passenger between the stations (o, d) appearing at the time t. Since the passengers are assumed to take a train and transfer so that the inconvenience to them will be minimized, we model the problem of routing them as the shortest path problem (refer to Hillier and Lieberman (2014) as well as Nemhauser and Wolsey (1999) on the shortest path problem).

We firstly construct a widely used train time-space graph (for instance, Kanai et al. (2011) call it a "passenger behavior network") from the originally planned train diagram and the set of time periods T. The vertices consists of the arrival/departure events of the trains at each station, and triplet $(o, d, t) \in S_{b<}^2 \times T$ for each direction $b \in B$. This means the appearance of passengers at the station o at the time t whose destination is d. The directed edges are drawn between vertices when the passengers can move between them directly: specifically, (i) an appearance of passengers to a train departure vertex from the same station, (ii) a train departure vertex to the train arrival vertex at the next station, (iii) a train arrival vertex to the train departure vertex at the same station, (iv) a train arrival vertex to a different train departure vertex at the same station. The edge (i) indicates a wait for the departure of the corresponding train, the edge (ii) travel by the train between the adjacent stations, the edge (iii) a wait on board at the station, and the edge (iv) a transfer between the two trains and a wait for the departure of the latter train. These form a directed acyclic graph.

We next set, for each edge, the amount of inconvenience caused by the corresponding activity. The waiting time multiplied by μ is imposed on (i), the traveling time on (ii), the dwell time on (iii), and the transfer penalty ν as well as the waiting time multiplied by μ on (iv). We then find a shortest path from each appearance vertex (o, d, t) to any train arrival vertex at d. We apply Dijkstra's algorithm on a directed acyclic graph (refer to Wolsey (1998) on the shortest path problem on a directed acyclic graph), and have the shortest path length, which is $A_t^{o,d}$.

3.5.2 Delay Spread Simulation

At Step 2, we input initial delay information α_r^s, δ_r^s. We temporarily set $\widehat{A}_r^s := \infty$ and $\widehat{D}_r^s := \infty$ to ignore the right-hand sides of the constraints (3.7) and (3.8). We also let $\widehat{\Delta}^A := \varnothing$ and $\widehat{\Delta}^D := \varnothing$ to ignore the constraints (3.11) and (3.12). We then decide a proper value of the large constant M_H concerning headway which is in the constraints (3.14)–(3.17) and (3.23)–(3.29) so that we can solve (*DMP*). These constraints are of the form

$$(\text{time of Event 1}) - (\text{time of Event 2}) \geq (0 \text{ or } I_*^*) - M_H \times (0,\ 1,\ 2 \text{ or } 3).$$

Overall, the constant satisfies

$$M_H \geq \max\{I_*^*\} - \min\{a_r^s, d_r^s\} + \max\{a_r^s, d_r^s\}.$$

Since a timetable in which all the trains are delayed for the maximum amount of initial delays is feasible, we let

$$M_H := \max\{I_*^*\} - \min\{A_r^s, D_r^s\} + (\max\{A_r^s, D_r^s\} + (\text{max. amount of initial delays})).$$

We let the values of the decision variables $\ell_{r,e}, u_r^{s,k}, g_{r_1,r_2}, x_{r_1,r_2}^s$ be those of the current timetable, and solve (*DMP*). Since the problem is a simple linear programming problem, it can be solved immediately by an optimization solver (refer to Hillier and Lieberman

(2014), Nemhauser and Wolsey (1999) and Williams (2013) on linear programming). Then we have a simulated timetable on the spread of the initial delay to other trains if we operate the trains in accordance with the current timetable.

3.5.3 Solving Arrival Delay Minimization Problem

We set the rescheduling start time h_0 at the beginning of Step 3, and let the arrival or departure time of the events which occurs before the rescheduling start time be its simulated time. We then solve (DMP) with the binary variables being unfixed and obtain a feasible rescheduled timetable as a reference point.

3.5.4 Solving Passenger Inconvenience Minimization Problem

At Step 4, we improve the rescheduled timetable in terms of passenger satisfaction. Firstly, we let $\widehat{A}_r^s, \widehat{D}_r^s$ be the rescheduled arrival/departure time in the optimal solution to (DMP) at Step 3. We next introduce the parameter I_{Flex} to widen the solution space for the passenger-oriented optimization. The right-hand sides of (3.7) and (3.8) in (IMP) are now decided, and we could make M_H smaller. We update

$$M_H := \min\{M_H,\ \max\{I_*^*\} - \min\{A_r^s, D_r^s\} + \max\{\widehat{A}_r^s, \widehat{D}_r^s\} + I_{\mathsf{Flex}}\}.$$

On the other hand, a proper value of the arbitrary large number M_I concerning inconvenience is still unknown. The constraints (3.37)–(3.39) involving M_I are of the form

$$\begin{aligned} M_I \geq\ & \text{(amount of inconvenience on (partial) Route 1)} \\ & - \text{(amount of inconvenience (partial) Route 2).} \end{aligned}$$

Consider here a route of a passenger on which his/her inconvenience will be maximized. When the waiting parameter value μ is bigger than one, the inconvenience will increase by waiting a train as long as possible. When $\mu < 1$, he/she will suffer more inconvenience by traveling on board as long as possible. The inconvenience to him/her will increase further if he/she transfers at every possible station and the transfer penalty ν is imposed. According to these observations, we decide the value of the arbitrary large number as follows:

$$M_I := \max\{\mu, 1\} \times (\max\{\widehat{A}_r^s, \widehat{D}_r^s\} + I_{\mathsf{Flex}} - \min\{A_r^s, D_r^s\}) + \nu \times |\max\{S_{\mathsf{Tra}(b,\,d)}\}|.$$

We finally solve (IMP) and obtain a desired rescheduled timetable.

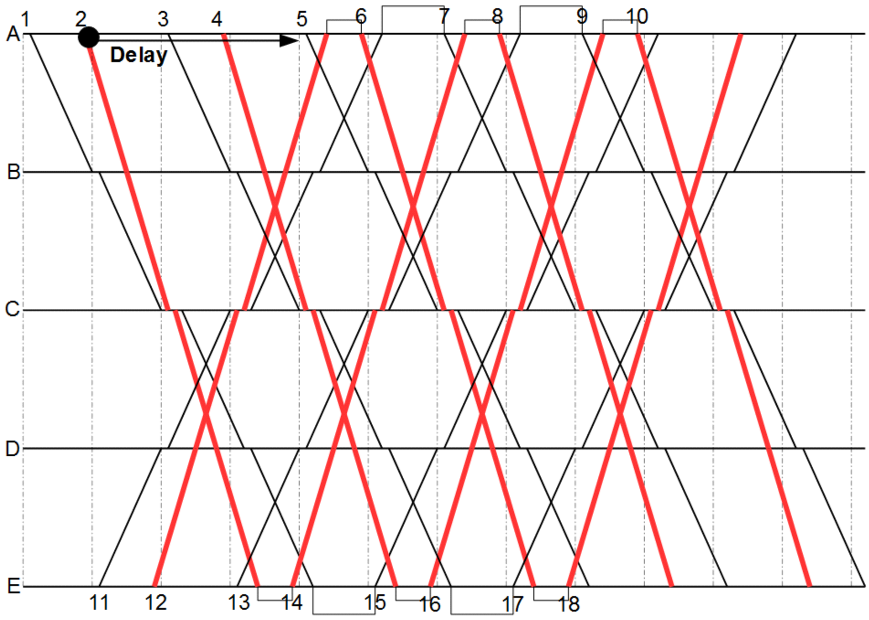

Fig. 3.11 Diagram of originally planned timetable and initial delay for arbitrary created example

3.6 Examples

We present rescheduled timetables proposed by our rescheduling algorithm for an arbitrary created timetable to observe how the solution to the arrival delay minimization problem (DMP) is improved in terms of the inconvenience when each arrival/departure event is allowed to be further delayed in the passenger inconvenience minimization problem (IMP). Figure 3.11 displays a diagram of the originally timetable and an initial delay. This example has five stations, ten westbound trains and eight eastbound trains. The vertical dashed line is depicted for every ten minutes. The timetable is cyclic in which its cycle time is 20 minutes. The bold and thin lines indicate express and local trains, respectively. The express trains stop at Stations A, C and E. These trains can overtake the local ones at any intermediate station though they do only at Station C in the originally planned timetable. Some of the trains turn back at the termini (e.g., Train 12 does at Station A and is next operated as Train 6). We deal with 569 different triplet (o, d, t) in the timetable, i.e., a group of passengers who appear at station o at time t and travel to station d. Each passenger appears at his/her origin station zero to three minutes before the departure of a train which he/she plans to take.

To this planned timetable, we input the delay of 30 minutes on the departure of Train 2 from Station A and assume that Trains 3 and 4 depart from a different track from a track which Train 2 depart from at Station A so that Trains 3 and 4 run on the originally planned time. We let the rescheduling design deadline and the rescheduling start time be also 30 minutes after the occurrence of the initial delay for simplicity, and then solve (DMP) and (IMP). We output incumbent solutions unless it terminates in 30 minutes. The computational environment described in the next section.

Figures 3.12, 3.13, 3.14 and 3.15 illustrate the diagram of the optimal solution to (DMP) as well as the diagrams of the optimal solutions to (IMP) with the inconvenience parameter values $(\mu, \nu) = (1, 1)$ concerning the expression (3.1) and the flexibility parameter $I_{\mathsf{Flex}} = 5, 10, 15, 20$ (in minutes). The inconvenience parameter value set means that each passenger takes a route which makes his/her traveling time be minimized and requires less number of transfers if the traveling time of two routes are equivalent, since he/she can transfer at one station and we let the minimum arrival interval between trains at any station be more than two minutes. The dashed lines in the diagrams indicate the originally planned timetable and the rolling stock assignment at termini.

The arrival delay minimization problem (DMP) is solved immediately. In Figure 3.12, Train 1 instead of the delayed Train 2 is next operated as Train 14 to reduce the spread of the delay. Train 13 stops longer and is overtaken at Station B since the delay does not affect its successor Train 9. The delay of Train 8 is decreased since its predecessor Train 14 overtakes Train 9. The most noticeable result is that Train 15 departs from its starting station later than Train 16 and that it is operated as an express train. The departure of Train 15 at Station E is much delayed since its predecessor is not Train 1 but Train 3, and the delay is recovered at Station B. This solution causes a long interval between the operation of two local Trains 13 and 17.

The passenger inconvenience minimization problem (IMP) with $(\mu, \nu) = (1, 1)$ and $I_{\mathsf{Flex}} = 5, 10, 15, 20$ are obtained before the rescheduling design deadline. In the optimal solution to (IMP) with $I_{\mathsf{Flex}} = 5$ shown in Figure 3.13 (we have the same solution when $I_{\mathsf{Flex}} = 10$), the type of Train 15 is local to reduce the inconvenience to the passengers from Station E to Stations B and D. Note that Train 10 is slightly delayed and that the connection between Trains 9 and 10 is kept by delaying the departure of Train 9 from Station C. The delay of Train 10 is spread by Train 16 and that of Train 16 is relevant to Train 15. These are observed on account of the flexibility parameter being still small; the departure time of local Train 15 from Station B has to be within five or ten minutes delay from the solution to (DMP), which is equivalent to the planned departure time, and it causes its early departure from Station E. This spread is resolved when we set $I_{\mathsf{Flex}} = 15$ displayed in Figure 3.14. Train 15 is intentionally delayed for over ten minutes and Train 16 is operated on time. In the rescheduled timetable when $I_{\mathsf{Flex}} = 20$, which is equal to the cycle time, Train 2 is virtually operated as Train 6 in the

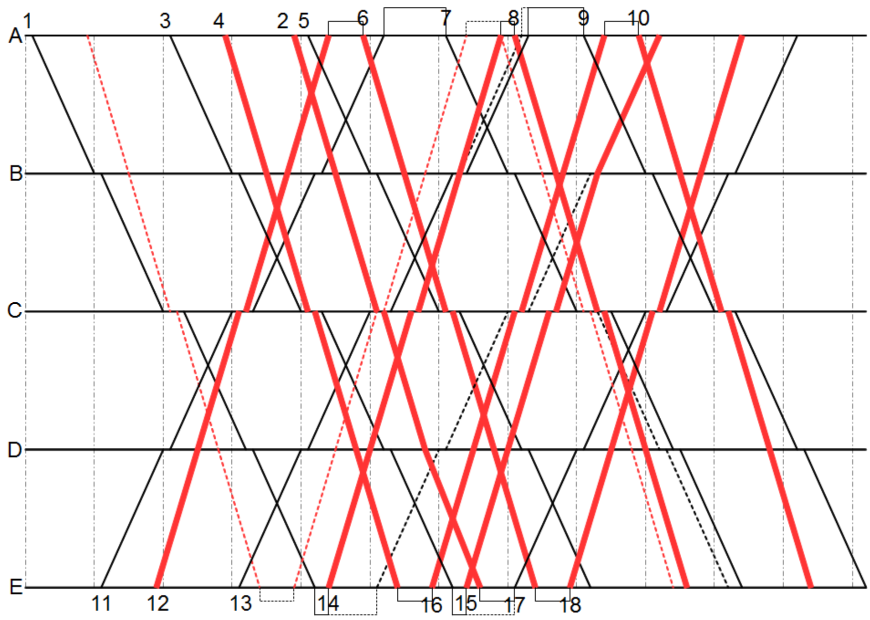

Fig. 3.12 Diagram of optimal solution to (DMP) for arbitrary created example

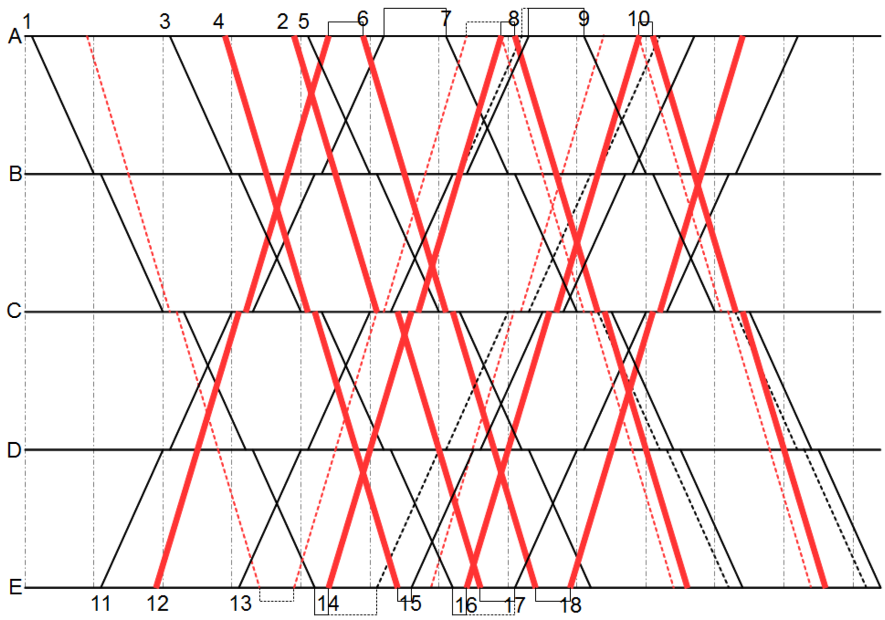

Fig. 3.13 Diagram of optimal solution to (IMP) with $(\mu, \nu) = (1, 1), I_{\mathsf{Flex}} = 5, 10$ for arbitrary created example

planned timetable. Similarly, other express Trains 6,8,14 and 16 are intentionally delayed for about 20 minutes in Figure 3.15. Trains 10 and 18 are operated as the local trains. We have this result since the originally planned timetable is optimal with respect to the minimization of the inconvenience to the passengers in an undisrupted situation (recall that each passenger appears at his/her origin station zero to three minutes before the

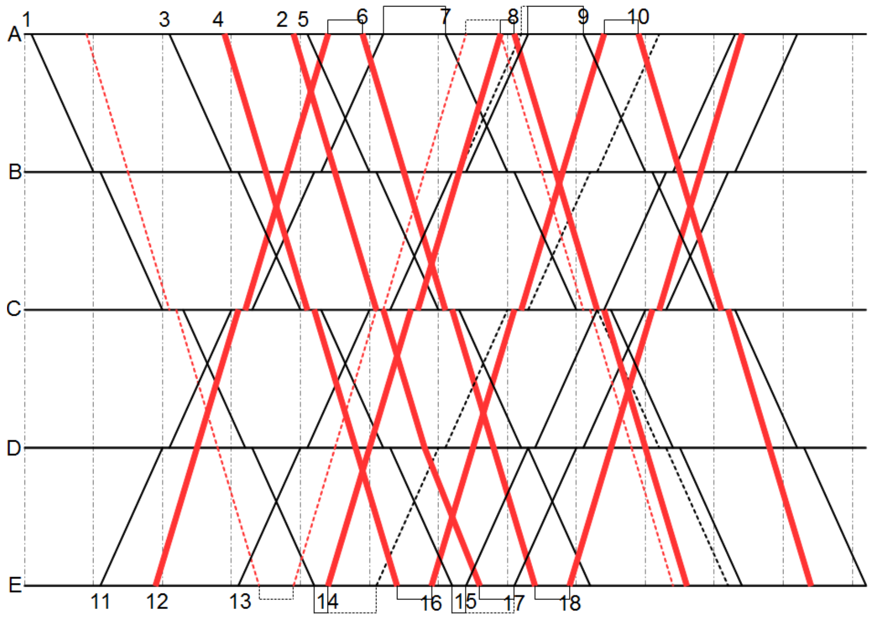

Fig. 3.14 Diagram of optimal solution to (IMP) with $(\mu, \nu) = (1, 1), I_{\mathsf{Flex}} = 15$ for arbitrary created example

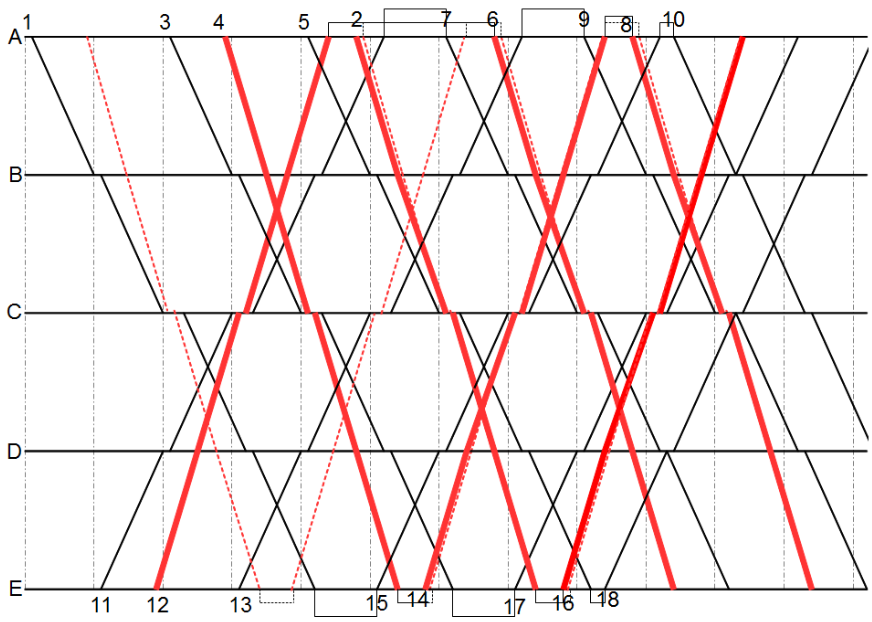

Fig. 3.15 Diagram of optimal solution to (IMP) with $(\mu, \nu) = (1, 1), I_{\mathsf{Flex}} = 20$ for arbitrary created example

departure of a train which he/she plans to take) and the flexibility parameter is large enough for the trains to be delayed for one cycle time. The retiming operation which is shown in Figure 3.6 and commonly performed by train dispatchers in Japan is also achieved by our inconvenience minimization formulation even though we do not explicitly model the rescheduling measure.

We present the solutions to (IMP) in Figures 3.16, 3.17 and 3.18 when we set $(\mu, \nu) = (2, 10)$ according to Railway Bureau, Ministry of Land, Infrastructure, Transport and Tourism (2012). The problem instances with $I_{\mathsf{Flex}} = 5, 10$ is solved to optimality while the ones with $I_{\mathsf{Flex}} = 15, 20$ are not before the rescheduling design deadline. We show the incumbent solution for these cases. We have the same solution for $I_{\mathsf{Flex}} = 5$ and 10. In this solution, Train 8 overtakes Train 7 at Station D while the change of the departure order is observed at at Station C when $(\mu, \nu) = (1, 1)$. The passengers who are taking Train 7 and is going to Station E do not transfer to Train 8 at Station C since the waiting and transfer penalties are heavier. Hence the local train is prioritized at Station C. Although the train is overtaken at Station D in the end, its arrival time at Station E is 103, one minute earlier than that for $(\mu, \nu) = (1, 1)$, to carry the passengers who do not transfer at Station C earlier. When $I_{\mathsf{Flex}} = 15$, we have a similar result to that for $(\mu, \nu) = (1, 1)$, except that Train 8 overtakes Train 7 at Station D. The incumbent solution for $I_{\mathsf{Flex}} = 20$ is the same as that when $(\mu, \nu) = (1, 1)$ since the originally planned timetable is also optimal when $(\mu, \nu) = (2, 10)$ for the distribution of the appearance of passengers.

3.7 Computational Results

3.7.1 Disruption Case and Computational Environment

We implement our rescheduling algorithm and assess its effectiveness for a 2012 weekend timetable of Chuo Line in Tokyo. Chuo Line has high frequency of train arrivals and departures which involve two or more types of trains in Japan, on which two express trains and five local trains are operated per 30 minutes on average. The passenger OD data are based on the ones investigated by Ministry of Land, Infrastructure, Transport and Tourism (2012) in 2010. Since they do not include time-dependent data, we assume that the number of passengers for each OD pair is uniformly distributed on time. Figure 3.19 shows the train operations between Tokyo and Tachikawa Stations, the main part of the whole line, for 50 minutes on the eastbound trains (From Tachikawa Station to Tokyo Station) and their successor eastbound trains in the daytime on the weekend, as well as an initial delay. The vertical dashed line is depicted for every ten minutes. The bold lines indicate express trains called Special Rapid Trains and the thin lines are Rapid Trains. Special Rapid Trains can overtake the other trains at Mitaka and Kokubunji Stations. We have omitted several minor stations which do not affect the order of the trains. The behavior of 2,108 different group of passengers (o, d, t) are considered in this timetable. Hence the instance is medium-sized.

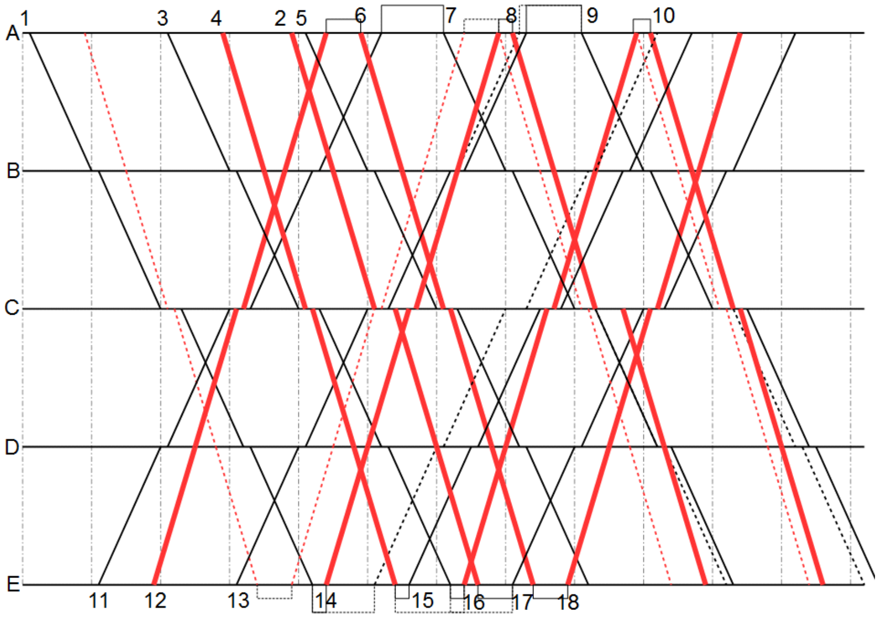

Fig. 3.16 Diagram of optimal solution to (IMP) with $(\mu, \nu) = (2, 10), I_{\mathsf{Flex}} = 5, 10$ for arbitrary created example

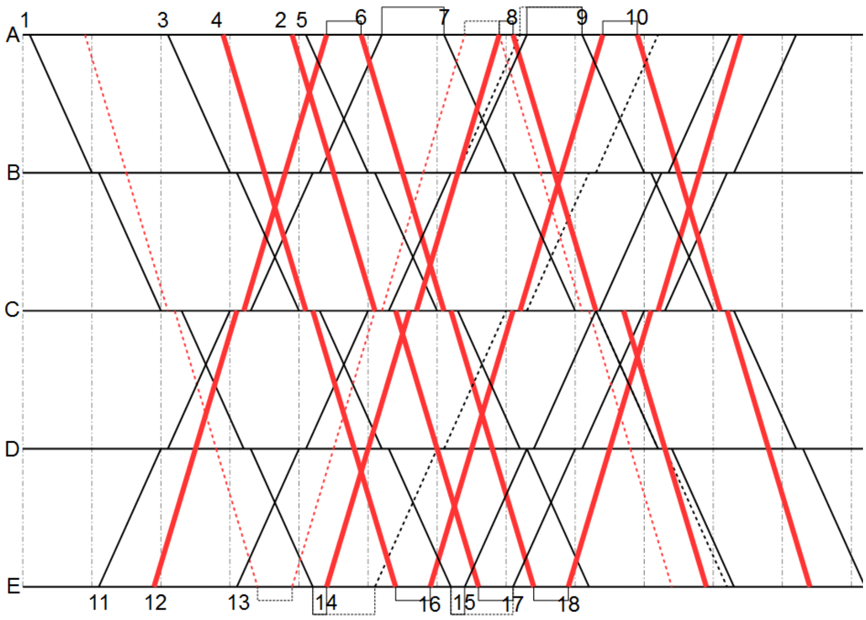

Fig. 3.17 Diagram of incumbent solution to (IMP) with $(\mu, \nu) = (2, 10), I_{\mathsf{Flex}} = 15$ for arbitrary created example

We input a delay of 15 minutes on the departure of Train 102 from Higashi-Koganei Station. We set $I_{\mathsf{Flex}} = 1, \dots, 5$ (in minutes). We let the rescheduling start time be 15 minutes after the occurrence of the delay. We then let the rescheduling design deadline be ten minutes after the occurrence of the delay and adopt incumbent solutions obtained

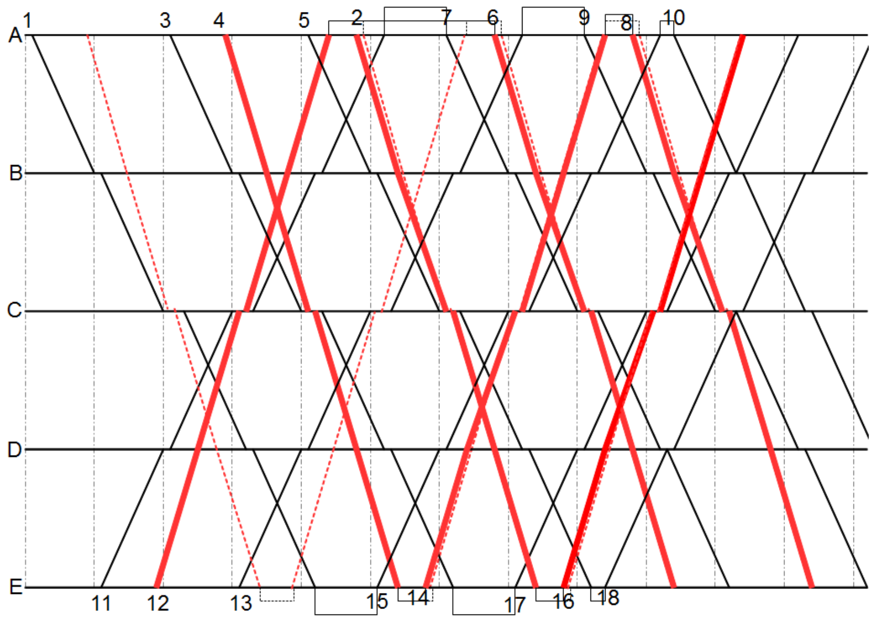

Fig. 3.18 Diagram of incumbent solution to (IMP) with $(\mu, \nu) = (2, 10)$, $I_{\mathsf{Flex}} = 20$ for arbitrary created example

before the time, assuming that the train dispatchers can input the modifications to the currents schedule proposed by the rescheduled solution to the train traffic control system and that the system transmits the information to a relevant staff. According to Technical Research Committee on Advanced Rail Transport Planning and Management, The Institute of Electrical Engineers of Japan (2010), advanced train traffic control systems have recently been being introduced to urban railway lines. These assumptions are hence valid. The combinations of the inconvenience parameters (μ, ν) are $(1, 1)$, and $(2, 10)$ according to Railway Bureau, Ministry of Land, Infrastructure, Transport and Tourism (2012). The former parameter value set means that each passenger takes a route which makes his/her traveling time be minimized and requires less number of transfers if the traveling time of two routes are equivalent, since he/she can transfer at two stations and the minimum arrival interval between trains at any station is more than two minutes. We solve the passenger inconvenience minimization problem (IMP) as well as the arrival delay minimization problem (DMP), and compare solutions with each other for various values of the timetable flexibility parameter I_{Flex}. The total inconvenience to passengers are decreased as I_{Flex} gets bigger since the solution space is enlarged. On the other hand, the total amount of delays may be increased; some trains may stop longer at a station to wait for extra passengers to come, for instance. The larger solution space also requires more computation time. The mixed integer programming problems are solved by Gurobi Optimizer 5.5.0 (the current version is offered by Gurobi Optimization (2016)) on a 64-bit

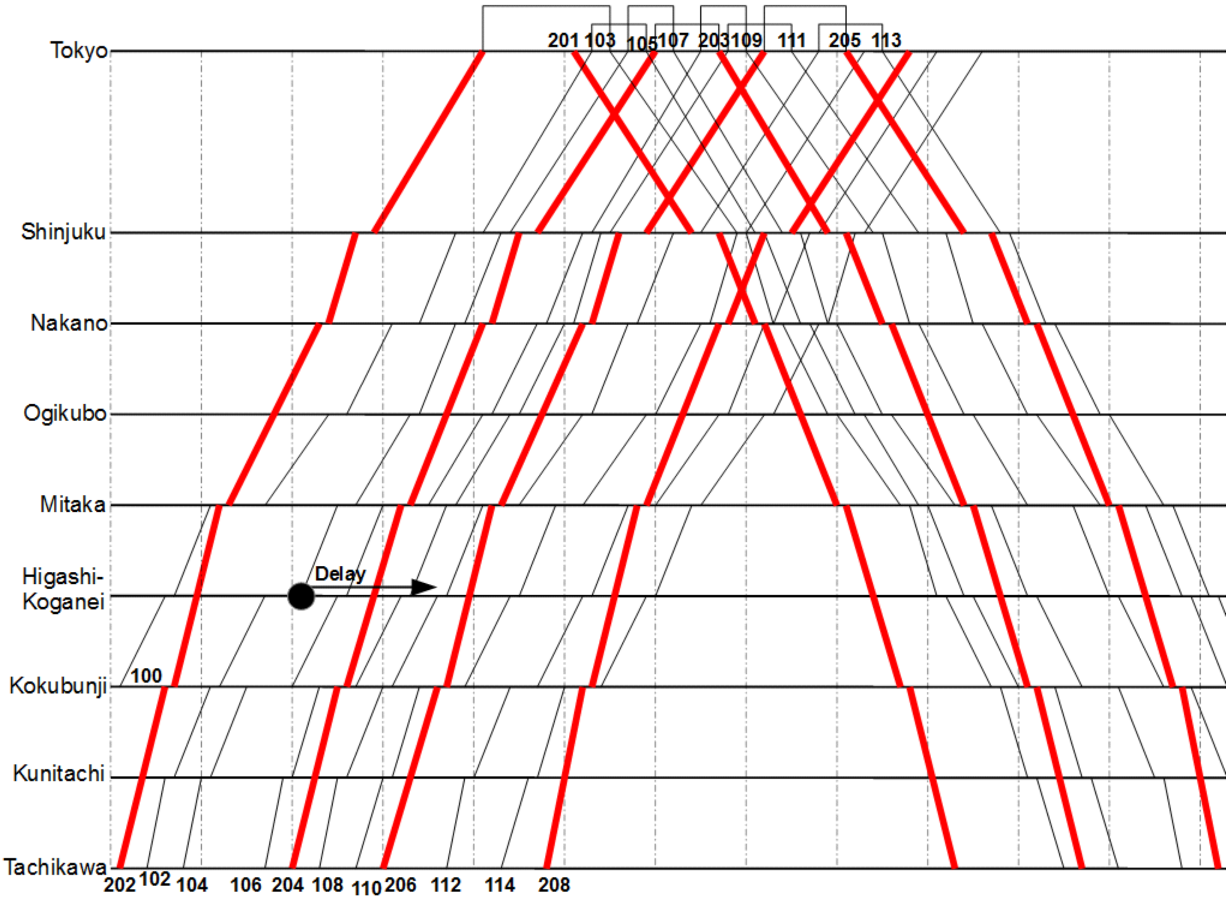

Fig. 3.19 Diagram of originally planned timetable and initial delay

Windows 7 PC having a Core i7-3930K CPU (six cores, twelve threads, 3.2–3.8 GHz) and 16 GB RAM. Six CPU threads are used by the Gurobi Optimizer. We still run the mixed integer programming solver after the rescheduling design deadline to observe the optimality of the solutions obtained before the deadline.

3.7.2 Rescheduling Results

The solutions to (DMP) and (IMP) with $(\mu, \nu) = (1, 1)$, $I_{\mathsf{Flex}} = 2, 3, 5$ (in minutes) are depicted in Figures 3.20, 3.21 and 3.22. Note that each of the solutions is optimal to the corresponding mixed integer problem which is found and provided to be optimal before the rescheduling design deadline except that the solutions to (IMP) with $I_{\mathsf{Flex}} = 5$ is found before this deadline while its optimality is provided after the time. We discuss the computation time in detail in the next subsection. The delay of Train 102 causes the consecutive delays of five Trains 104, 106, 204, 108 and 206 traveling in the same direction and four of their successor Trains 107, 203, 109 and 111 for all the solutions.

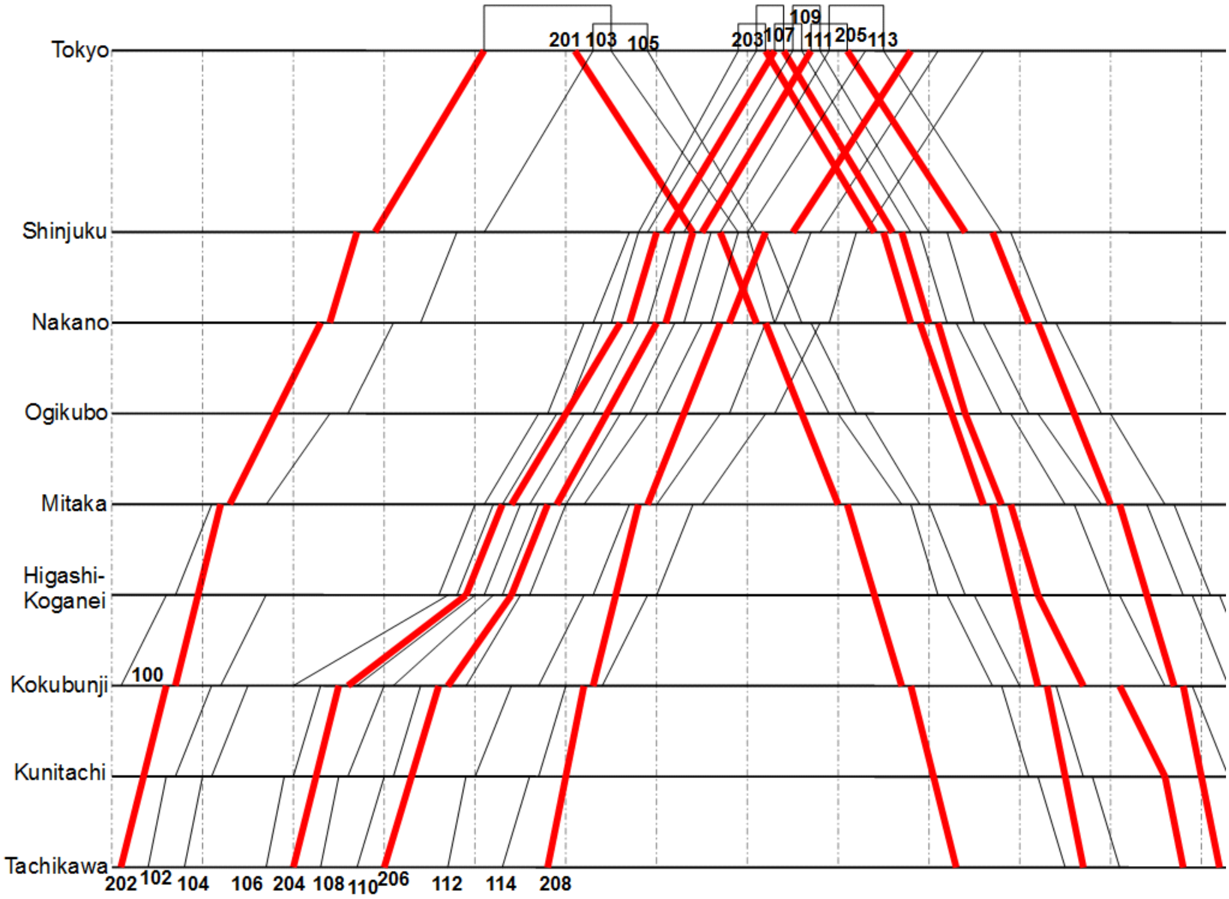

Fig. 3.20 Diagram of optimal solution to (DMP)

In the optimal solution to (DMP), Rapid Train 104 is not overtaken by Special Rapid Train 204 at Mitaka, although the latter has a higher priority and more passengers are on it. The delay of Train 107, which is the successor of Train 104, is recovered by operating it as Special Rapid Train. We solve (IMP) with $I_{\mathsf{Flex}} = 1$ and have the same departing order as that in the solution to (DMP). Several of the arrival and departure times of the trains are changed. For $I_{\mathsf{Flex}} = 2$ and $I_{\mathsf{Flex}} = 3$ (we have the same solution for the both parameter values), Train 204 overtakes Train 104. The departures of Trains 201, 103 and 105 from Tokyo are slightly delayed and the interval between the departures of Trains 105 and 203 is still over ten minutes due to the small parameter value. We have a similar result for $I_{\mathsf{Flex}} = 4$. More equal intervals are observed when we set $I_{\mathsf{Flex}} = 5$ since we let the number of passengers for each OD pair be uniformly distributed on time. This resembles the retiming operation which is shown in Figure 3.7 and commonly performed by train dispatchers in Japan though we do not consider the train congestion into account. Note that we do not explicitly model the rescheduling measure. On the other hand, there are still two successive Special Rapid Trains departing from Tokyo. The departure time

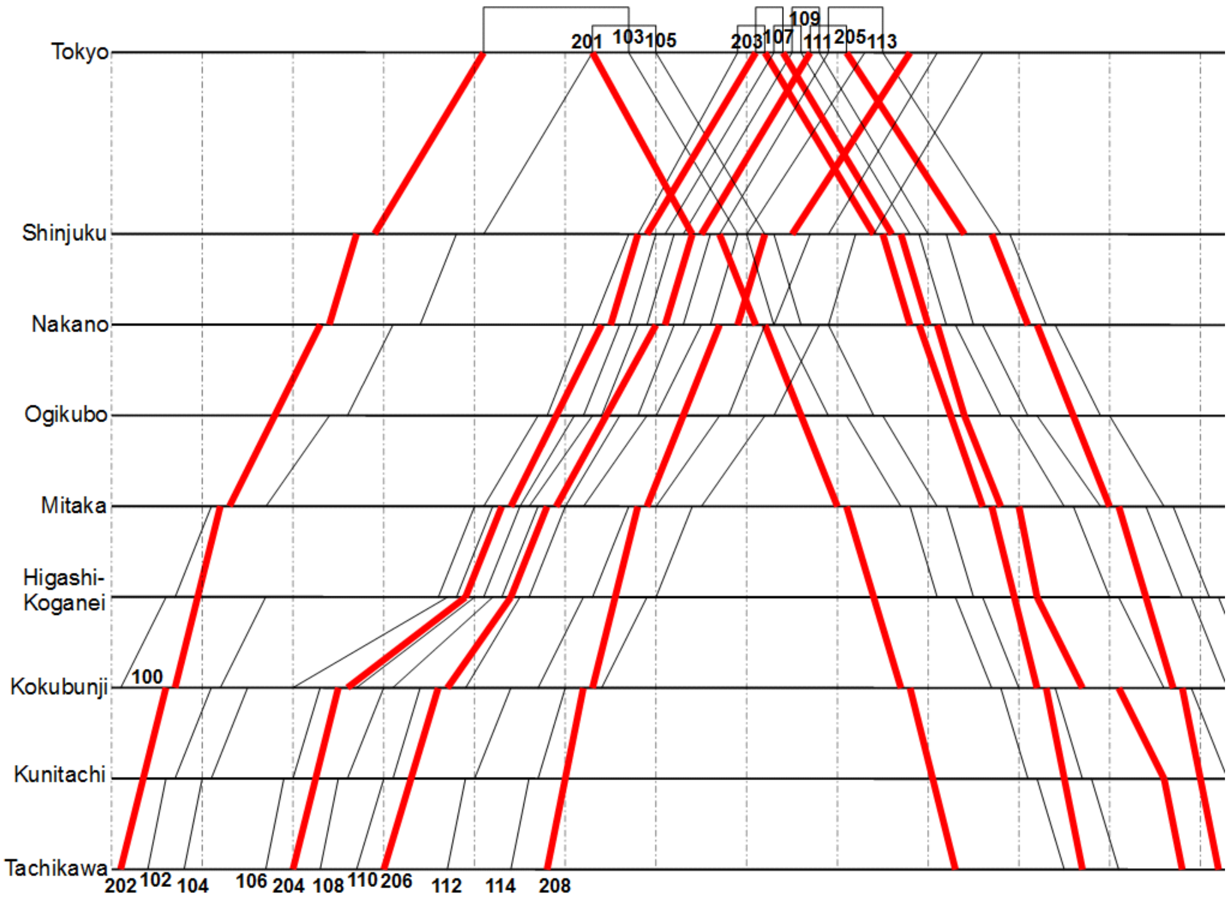

Fig. 3.21 Diagram of optimal solution to (IMP) with $(\mu, \nu) = (1, 1), I_{\mathsf{Flex}} = 2, 3$

of Train 107 from Tokyo is the same in the optimal solutions to (DMP) and (IMP) with $I_{\mathsf{Flex}} = 5$. It cannot be earlier due to the congested traffic around the crossing at the station. In the solution to (DMP), the Special Rapid Train runs from Tokyo to Higashi-Koganei in 28 minutes. When any Rapid Train runs between the stations as fast as it can, it will take 34 minutes at a minimum due to the minimum running time between each pair of stations and the minimum dwell time at every station. If Train 107 is operated as Rapid Train, the difference of the departure time from Higashi-Koganei will be six minutes or more. This exceeds the value of I_{Flex} and hence the train cannot be Rapid Train.

The solutions to (IMP) when we set $(\mu, \nu) = (2, 10)$ are displayed in Figures 3.23 and 3.24. They are found and proved to be optimal before the rescheduling design deadline. The diagram for $I_{\mathsf{Flex}} = 1$ is almost the same result as (DMP). The difference is that Train 105 is slightly delayed. When $I_{\mathsf{Flex}} = 2, \ldots, 5$, the solutions are equivalent. The diagram is not so different from that for $I_{\mathsf{Flex}} = 1$. Train 105 is delayed for one more minute between Nakano and Kokubunji to equalize the intervals among Train 103, Train 105 and the succeeding trains. It is, however, limited due to the arrival headway between Train 105

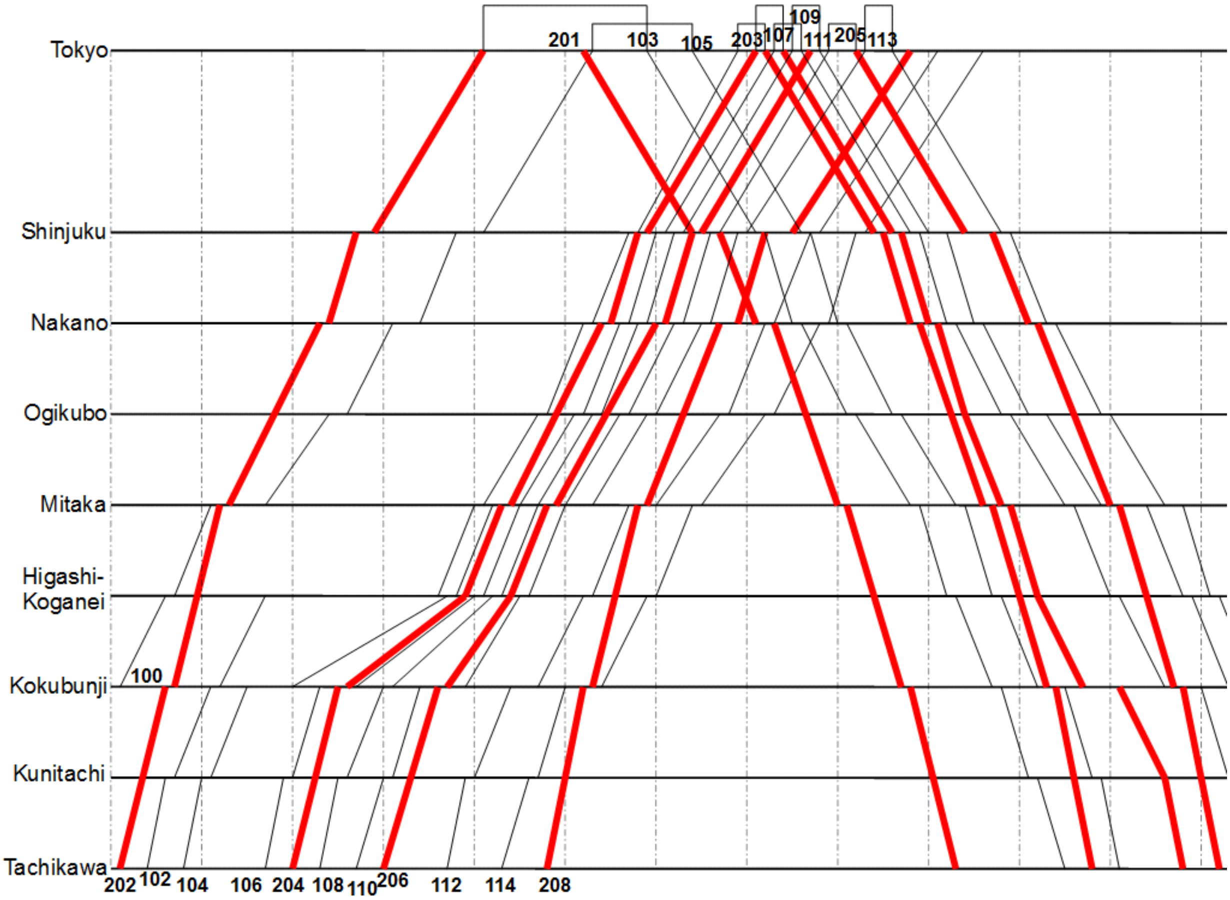

Fig. 3.22 Diagram of optimal solution to (IMP) with $(\mu, \nu) = (1, 1), I_{\mathsf{Flex}} = 5$

and Train 203 at Kokubunji. Train 204 does not overtake Train 104 at Mitaka, which is a different result from the solution when $(\mu, \nu) = (1, 1)$ and $I_{\mathsf{Flex}} = 2$. If the passengers who is taking Train 104 transfer to Train 204 at the station, they will reach Nakano, Shinjuku or Tokyo in at most two minutes earlier. This is smaller than the value of $\nu = 10$.

Table 3.3 shows the objective function values of (DMP) and (IMP) for each formulation, combination of the inconvenience parameter values, flexibility parameter value I_{Flex} (in minutes) and pair of the large numbers M_H, M_I (in minutes) in the formulation. The computation time is also displayed (in seconds). We let "found" be the elapsed time until the final incumbent solution is found, and "proved" be the time until the solution is proved to be optimal and then the algorithm stops. Concerning the objective values, we observe the trade-off between the total amount of delay and the total amount of increased inconvenience on the whole. We have a better solution in terms of the inconvenience at the expense of the minimum delay timetable by introducing flexibility in the timetable. When the optimal order of the trains in terms the inconvenience is different from that in the delay-minimized timetable, there are more chances of their reordering if we have more

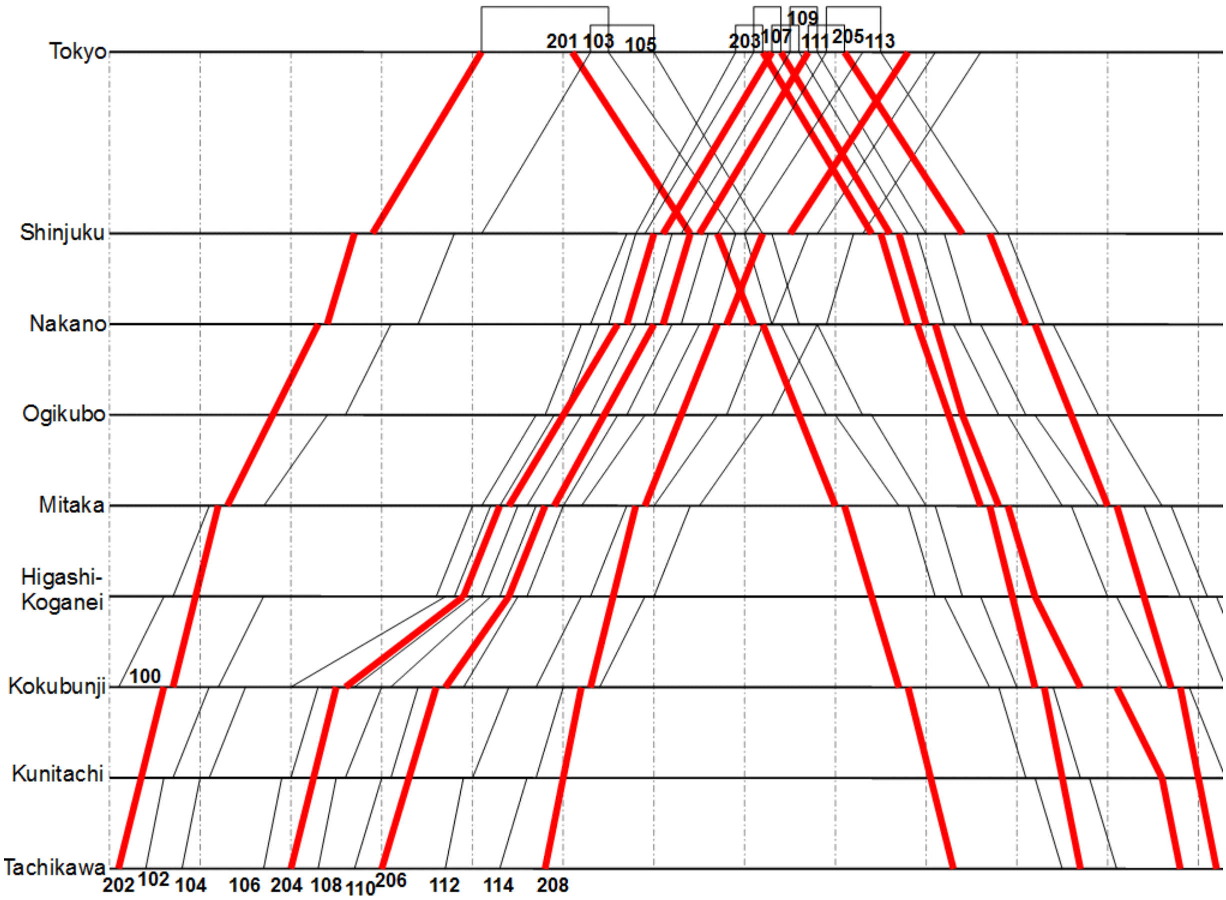

Fig. 3.23 Diagram of optimal solution to (IMP) with $(\mu, \nu) = (2, 10), I_{\mathsf{Flex}} = 1$

flexibility in arrival and departure times of the trains.

3.7.3 Computation Time

We next discuss the computation time required to solve the instances, which is displayed in Table 3.3. Note that the computation time to solve (IMP) displayed in the table does not include the time to solve (DMP), although the delay-minimized solutions are needed as a part of the input of (IMP). In this cases, (DMP) is solved to optimality immediately, i.e., a feasible rescheduled timetable as a reference point is obtained in a short time. The trade-off between obtaining a better solution in terms of the inconvenience by enlarging the solution space and the computation time is observed in the table. The computation time to solve (IMP) grows dramatically when the flexibility parameter value exceeds a certain threshold for each case and pair of the inconvenience parameters. The output solution to (IMP) with $I_{\mathsf{Flex}} = 5$ and $(\mu, \nu) = (1, 1)$ is not proved to be optimal before the rescheduling design deadline, which is ten minutes (600 seconds) after the occurrence

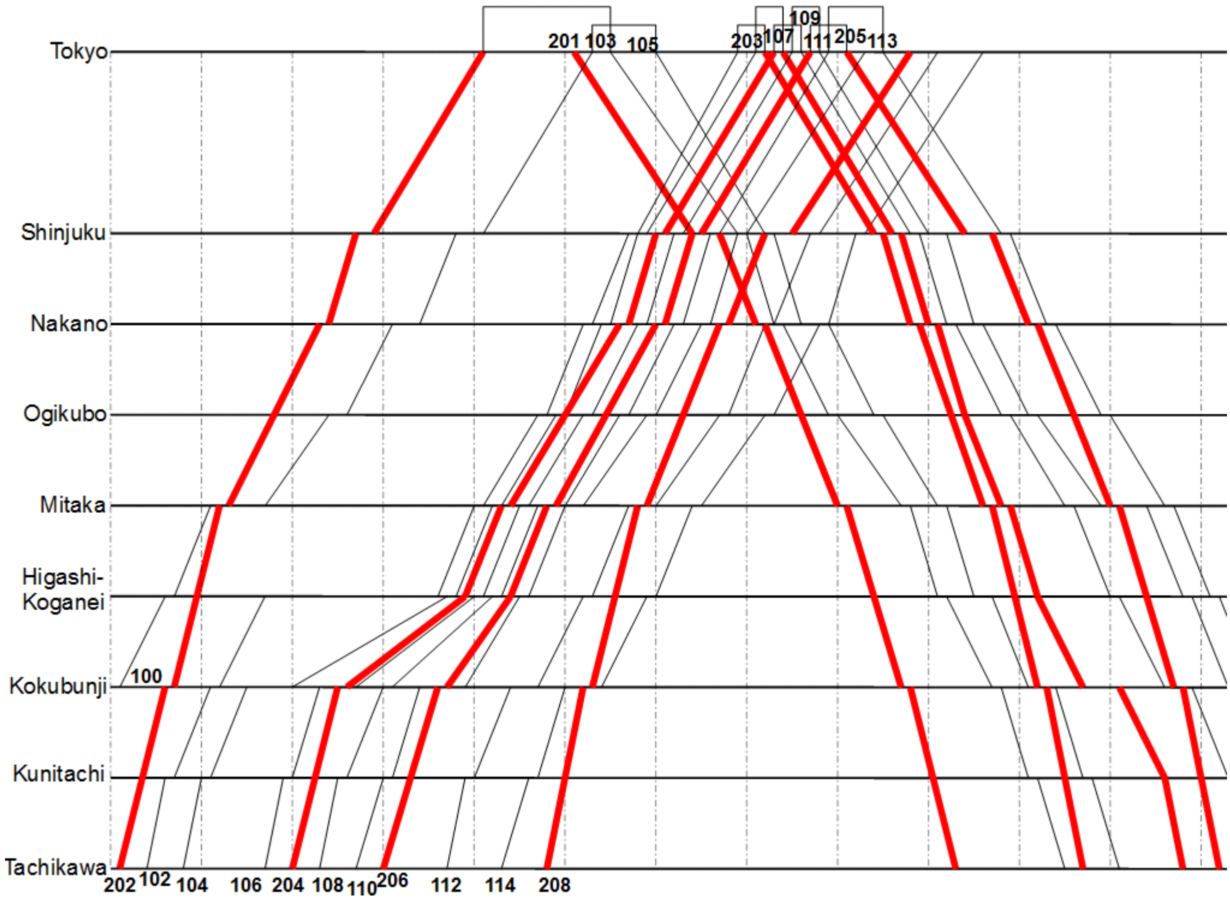

Fig. 3.24 Diagram of optimal solution to (IMP) with $(\mu, \nu) = (2, 10)$, $I_{\mathsf{Flex}} = 2, \ldots, 5$

of the delay, whereas the best incumbent solution is found before the deadline and it is optimal in fact. A large part of the whole computation is spent on confirming the optimality of the solution. We have $z^{o,d}_{t,r}$ for different $(o, d, t, r) \in S^2_{b<} \times T \times R_b$ and the lower bounds obtained by the linear programming relaxation problems are not sufficiently strong from the constraints (3.31), (3.39) and (3.40) as well as the objective function. In spite that the bigger inconvenience parameters require larger values of M_I and that it makes the linear relaxation weaker, the problems with $(\mu, \nu) = (2, 10)$ are generally solved in less time. Without the additional redundant constraints (3.41)–(3.47), the proof of optimality in solving (IMP) is intractable even when $I_{\mathsf{Flex}} = 1$ since the constraints fix most of $z^{o,d}_{t,r}$ when $\ell_{r,e}$ and $x^s_{r_1,r_2}$ take binary values in the branch-and-cut process.

We also assess the computation time depending on the amount of an initial delay. The required time for us to solve (DMP) and (IMP) with $(\mu, \nu) = (1, 1)$ with delays of 15, 20 or 25 minutes are described in Table 3.4. We let the rescheduling start time be 15, 20 or 25 minutes after the occurrence of the delay depending on a case, and the rescheduling design deadlines be five minutes before the rescheduling start time. We still run the

Table 3.3 Objective values and CPU time

Problem	(μ, ν)	I_{Flex} (m)	M_H (m)	M_I (m)	Objective value (DMP) (m)	Objective value (IMP) (•)	Time (s) Found	Time (s) Proved
(DMP)	–	–	156	–	380	233,010	2.0	3.0
(IMP)	(1, 1)	1	142	141	383	225,350	3.0	3.3
		2	143	142	388	219,120	12.0	13.3
		3	144	143	388	219,120	19.0	19.9
		4	145	144	397	218,480	26.0	93.8
		5	146	145	417	217,720	247.0	1,722.0
	(2,10)	1	142	298	383	344,260	2.0	3.1
		2	143	300	387	341,870	11.0	11.4
		3	144	302	387	341,870	13.0	14.0
		4	145	304	387	341,870	93.0	93.7
		5	146	306	387	341,870	164.0	286.0

• The increased amount of inconvenience times the number of passengers is a unit.

mixed integer programming solver after the rescheduling design deadlines to observe the optimality of the solutions obtained before the deadlines, and stop it after 1,800 seconds (30 minutes). For the cases where the best incumbent solutions are not proven to be optimal in 1,800 seconds, the MIP gap defined by ((objective value of incumbent solution) − (lower bound))/(lower bound) × 100 is described in the table.

The last Train 113 runs on time in the solutions to (DMP) even in the case of the delay of 25 minutes, and therefore the values of M_H, M_I are the same when we solve (IMP). The computation time required to solve (DMP) grows dramatically as the amount of initial delays increases, and the overall rescheduling procedure also does. Although (IMP) itself for $I_{\mathsf{Flex}} = 1, \ldots 3$ is solved in real time, we should note that the solution space has been narrowed by solving (DMP) in advance. The inconvenience minimization does not terminate in 1,800 seconds when the amount of delay is 20 or 25 minutes and $I_{\mathsf{Flex}} \geq 4$. If we add the computation time to prove the optimality of the solutions to (DMP) to the time to find the last incumbent solutions to (IMP), we can say that the last incumbent solutions are obtained before the rescheduling design deadlines except for the case with the 25 minutes delay and $I_{\mathsf{Flex}} = 5$. The optimality gaps are small and there are possibilities that the solutions are optimal.

Recall that this instance is a contracted railway line from the actual one with heavy traffic by omitting the several stations. We solve (DMP) for this medium-sized instance and try to improve the rescheduled timetable in terms of the inconvenience to passengers by spending additional computation time solving (IMP). The computation could last until the rescheduling design deadline comes.

Table 3.4 Amount of initial delay and CPU time with $(\mu, \nu) = (1, 1)$

Amount of delay (m)	Problem	I_{Flex} (m)	M_H (m)	M_I (m)	Gap (%)	Time (s) Found	Time (s) Proved
15	(DMP)	–	156	–	0.00	2.0	3.0
	(IMP)	1	142	141	0.00	3.0	3.3
		2	143	142	0.00	12.0	13.3
		3	144	143	0.00	19.0	19.9
		4	145	144	0.00	26.0	93.8
		5	146	145	0.00	247.0	1,722.0
20	(DMP)	–	161	–	0.00	43.0	44.7
	(IMP)	1	142	141	0.00	3.0	3.8
		2	143	142	0.00	16.0	20.4
		3	144	143	0.00	46.0	50.3
		4	145	144	1.11	843.0[a]	–[b]
		5	146	145	1.46	821.0[a]	–[b]
25	(DMP)	–	166	–	0.00	192.0	335.7
	(IMP)	1	142	141	0.00	2.0	2.7
		2	143	142	0.00	21.0	41.3
		3	144	143	0.00	86.0	153.7
		4	145	144	0.49	188.0[a]	–[b]
		5	146	145	0.61	878.0[a]	–[b]

[a] Last incumbent solution is found after this elapsed time.
[b] Incumbent solution is not proved to be optimal in 1,800 seconds.

3.8 Conclusions

We have discussed in this chapter the timetable rescheduling of passenger trains while railway operations on a line are disrupted by disturbance. We have focused on the situations where all of the trains are due to be and can be operated between the ends of the whole line or a divided subset of the line, and require detailed rescheduling measures to manage the delay, including a case where general rescheduling measures have already been performed. We have defined the timetable rescheduling as to manage the delays of a given set of trains by applying a series of detailed rescheduling measures so that further inconvenience to passengers should be minimized. Various kinds of constraints on the train operations has to be satisfied so that the trains could be operated in accordance with the rescheduled timetable. A series of constraints on the passenger's behavior has also to be satisfied so that his/her route should be rational. A new timetable from the rescheduling start time has to be prepared before the rescheduling design deadline comes.

We have formulated the timetable rescheduling problem as two mixed integer programming problems. We have let events in the train operations be arrivals or departures of trains, and have changed their event time. Among detailed rescheduling measures, the reordering, the local rerouting and the retiming of trains are formulated, in addition to the change of a train type and the change of rolling stock assignment. We have let the passenger rerouting be possible and the passengers may transfer once or more on their route. The train operations and the passengers' behavior are simultaneously modeled. We have also let the inconvenience to each passenger consist of his/her traveling time on board, waiting time at platforms and the number of transfers. We have first modeled and solved the delay minimization problem. We have next introduced some flexibility in the delay-minimized timetable and solved the inconvenience minimization problem.

Numerical results based on the medium-sized railway line and delay contracted from the actual line with heavy train traffic have indicated the trade-off between the total amount of delay and the total amount of increased inconvenience. We have obtained a better solution in terms of the inconvenience at the expense of a minimum delay timetable by introducing flexibility in the delay-minimized timetable. The results have also indicated the other trade-off between obtaining a better solution by enlarging the solution space and the computation time. The computation could last until the rescheduling design deadline comes.

Chapter 4

Rolling Stock Rescheduling

4.1 Introduction

4.1.1 Background

In this chapter, we present rolling stock rescheduling of passenger or freight trains while railway operations on a line or a network are disrupted by disturbance, based on (K.) Sato and Fukumura (2012). Trains, or precisely a train timetable, cannot be realized unless rolling stock is assigned to each of the trains. The minimum unit of rolling stock that can be assigned to a train is called a rolling stock unit. Each rolling stock unit has been assigned to its planned sequence of trains whose assigned section and time are specified. The position of a unit should also be stated when a train is composed of two or more units. A periodic inspection (or called maintenance) of each unit, for every a few or several days, has to be carried out. It takes a few hours. In the rolling stock schedule, it is also specified when and where every inspection is done. The inspection is done at a rolling stock depot or a station which has facilities for it. The rolling stock rescheduling is carried out, at a certain point of time, when the railway operations in accordance with the current rolling stock schedule are being, or those to be implemented from the time to the following hours will be, delayed to some extent or unable to be carried out, owing to the disturbance.

Disruption to the rolling stock schedule is caused by disturbance which primarily affect the rolling stock schedule, a malfunction of a component of a rolling stock unit for instance, and the rescheduled timetable. Examples of the last case is as follows. There is a case, mainly at a terminus of a line in Japan, where the next planned task of a rolling stock unit which is delayed and has not yet arrived at the station, is assigned to another unit which is being parked at the station temporarily, to prevent the delay from spreading. This operation is classified as a timetable rescheduling measure in a broad sense in spite of the rolling stock reassignment, since it is relevant to recovery of train delay. Even though the delay is recovered in this case, the rolling stock schedule is in turn disrupted.

The rolling stock rescheduling plan has to be designed and implemented in real-time, since there is a case where any rolling stock unit has not yet been assigned to trains which will depart after the following minutes. Another case is that the disrupted situation might change while we are considering a rescheduling plan and that the plan might not be applicable in the latest situation. The dispatchers in charge of the rolling stock rescheduling have to communicate the modified pieces from the current schedule in the new schedule to train crews and a rolling stock depot staff via a station staff, by radio, by phone or by any information technology. This task is cumbersome, hence the scale of the actions to be rescheduled should be as small as possible. Another difficulty in the rolling stock rescheduling is the management of the periodic inspections. We have to ensure that inspections are also properly carried out in a disrupted and rescheduled sequence of trains assigned to each unit. An unscheduled inspection can be carried out, though it is not desirable since it will be a burden on the rolling stock depot staff and the number of units that we can simultaneously inspect is limited at each depot and at any point of time. We have to make the rescheduled plan operator-oriented by taking these requests into consideration.

4.1.2 Terminology

On the basis of the disruption management context described above, we define terminology of the rolling stock rescheduling on a railway line or a railway network as follows. We call a certain point of time at which we have noticed necessity for the rescheduling the *current time*. A *disrupted situation* is train and rolling stock operations at the current time which is being or will be disrupted by disturbance. A certain amount of time is required to plan a new schedule and communicate necessary orders to a rolling stock depot staff, train crews and staff members involved in shunting at stations. If part of the rolling stock operations is managed by a certain system (a train traffic control system for instance), the schedule is input to it. We let a *rescheduling design deadline* be a deadline for planning the new schedule. The time when the communication and the input is completed and we can implement the schedule is called *rescheduling start time*. We also set the *rescheduling period*, which is hours from the rescheduling start time.

A timetable, originally planned in its scheduling phase or rescheduled once or more before the current time in any manner, is given. We call it the *current timetable*. The *current rolling stock schedule* on the line or the network is also given, which is originally planned or rescheduled once or more before the current time. An *updated timetable* is obtained at the current time, which can be equivalent to the current one, be further disrupted by the disturbance or be rescheduled in timetable rescheduling. The minimum unit of rolling stock that can be assigned to a train, which cannot be uncoupled any more or is not done due to a certain operational rule, is called a *rolling stock unit.* A rolling stock unit is involved in the rescheduling if it is or will be on operated in the rescheduling period in the current rolling stock schedule. A *reserve rolling stock unit* may be available. Such a unit is at a *rolling stock depot.*

In any rolling stock schedule, each rolling stock unit is assigned to its planned sequence of trains whose assigned section and time are specified. We call it a *sequence of train tasks of a rolling stock unit* and the sequence in the current rolling stock schedule the *current sequence of train tasks of a rolling stock unit.* A *position* of a unit should also be stated when a train is composed of two or more units. A sequence of tasks of a unit may be arbitrarily divided into several units and each of them is called a *duty.* A duty typically means a sequence of tasks for one day. Each unit will be ready for the rescheduling at a certain time and place after the rescheduling start time. If it is a reserve rolling stock unit, it will be ready at the rescheduling start time and at the depot. The trains in the current sequence of train tasks of each rolling stock unit can be reassigned after the time for the unit be ready for the rescheduling. We call each of them a *train task.* Trains in the rescheduling period to which any unit has not yet been assigned to are included in the train tasks (they may be divided into several tasks at certain stations) and canceled trains are excluded. If the train of a train task is an out-of-service train, it is called a *deadhead train task.* The first train task in the current sequence of each unit after the rescheduling period expires is called a *convergence task.* Any unit needs *preparation time* from the end of each train task to the beginning of the next one. An *importance* value is assigned to each train task, e.g., the importance of deadhead train tasks is zero and that of the rest of tasks depends on the priority of the train.

Every *inspection* is also specified in the sequence of train tasks of a rolling stock unit. It takes a few hours. The word “inspection” is widely used in Japan and it is called “maintenance” in some countries. A periodic inspection of each unit, for every a few or several days, has to be carried out. We call it the *inspection constraint.* It is carried out at a rolling stock depot or a station which has facilities for it. The maximum interval allowed between two consecutive inspections for a unit is called an *inspection interval.* It ordinarily depends on the type of a rolling stock unit. The number of units that we can simultaneously inspect is limited at each depot and at any point of time. We call it an *inspection capacity.*

4.1.3 Approach

We let an updated timetable, the current rolling stock schedule, the disrupted situation, the rescheduling design deadline and the rescheduling start time be given on a railway line or a railway network of passenger or freight trains. The *rolling stock rescheduling* in this chapter is then to reassign the rolling stock units the train tasks so that the scale of the actions to be rescheduled should be as small as possible and that unscheduled inspections should be as few as possible. Two consecutive inspections of each of the units must not violate the maximum inspection interval permitted. Each train is assumed to consist of one unit, which is common on many lines and networks, or to be composed of two or more units which can be freely coupled. Routing inside depots and stations as well as coupling and uncoupling of units are discussed in shunting rescheduling. A deadhead train tasks can be canceled or up to the specified number of coupled units. A rolling stock unit is not permitted to be assigned to a train task if its type is not suitable for the characteristics of the train. The preparation time is necessary between two consecutive train operations to which the same unit is assigned. The inspection capacity exists at each rolling stock depot or station at each time span. We have to reschedule the rolling stock assignment to the tasks involved in the rescheduling period before the rescheduling design deadline comes.

If the rolling stock rescheduling has failed, i.e., we cannot reassign all the train tasks to the rolling stock units, timetable rescheduling (which is not restricted to the methods discussed in our timetable rescheduling) have to be carried out so that the situation should be resolved. Before that, it is useful to clarify which tasks are unassigned. If we can choose unassigned tasks in the overall train task set, it is desirable that they are of less importance since they will be delayed or canceled in the timetable rescheduling. We call the detection of unassigned train tasks which are less important the *rolling stock uncovered train task detection*.

Firstly we construct the *rolling stock rescheduling network* from the rescheduling situation. We then formulate the rolling stock rescheduling as an integer programming problem which can be considered as a variant of the set partitioning problem. We solve it by column generation since all the possible sequence of tasks of the rolling stock units are not known in advance and it will take large amount of time to enumerate them. The columns generated in the algorithm is not enough for us to find an optimal solution to the original integer programming problem. Hence our approach provides near-optimal solutions in general. If no solution is found, we formulate the rolling stock uncovered train task detection as another integer programming problem which can be seen as a variant of the set packing problem. We also solve the problem by column generation. Large problem instances of the set partitioning problem are known to be difficult to solve, and therefore we temporarily introduce relaxation of the rolling stock rescheduling named *set-covering relaxation* to enhance the computation. The column generation subproblems with the periodic rolling stock unit inspection constraint are reduced to the shortest path problems which can be solved in polynomial time and inspections are carried out once or more in a rescheduled sequence of train tasks for each rolling stock unit if necessary. The readers are recommended referring to Hillier and Lieberman (2014) as well as Nemhauser and Wolsey (1999) on a network, Hillier and Lieberman (2014), Nemhauser and Wolsey (1999), Williams (2013) and Wolsey (1998) on integer programming, Nemhauser and Wolsey (1999) on the set partitioning problem and the set packing problem as well as the set covering problem and Desrosiers and Lübbecke (2005) as well as Wolsey (1998) on column generation. Barnhart et al. (1998), Lübbecke and Desrosiers (2005) and Vanderbeck (2005) discuss advanced topics on column generation. Other topics on operations research and mathematical optimization are also expounded by Hillier and Lieberman (2014), Nemhauser and Wolsey (1999) and Wolsey (1998).

4.2 Literature Review

There has been a smaller number of rolling stock rescheduling studies than that of timetable rescheduling studies in the rail transport literature. Adenso-Díaz et al. (1999), Cadarso et al. (2013), Fekete et al. (2011), Flier et al. (2008), Kunimatsu and Hirai (2009), Tomii et al. (2005) and Veelenturf et al. (2011) discuss timetable rescheduling and rolling stock assignment at the same time. Note that these studies are aimed at managing the timetable disruption.

Hara et al. (2004) presents a field trial of a rolling stock support system, whose rescheduling algorithm is not described in detail. Louwerse (2009) formulates rolling stock rescheduling as an integer programming problem which minimizes the sum of seat-shortages with respect to an expected demand and the differences between the new rolling stock allocation and the standard rolling stock allocation. Hirai (2010, Chapter 6) introduces a concept called "train trading," which means that rolling stock units are considered as an agent and they exchange train tasks in the central market. An integer multicommodity network flow model which can be applied to rolling stock rescheduling or crew rescheduling is provided by (T.) Sato et al. (2009), and it is solved by partial path exchange heuristics and a local search. Almost the same problem is discussed by (T.) Sato et al. (2012) and it is solved by Lagrangian relaxation. In these models, inspection or maintenance of rolling stock units is not mentioned. Shunting of rolling stock units is not, either.

Jespersen-Groth et al. (2009b) and Nielsen (2011, Chapter 3) presents complex integer programming formulations based on a flow of rolling stock units which optimize the rescheduling in terms of several perspectives including seat-shortages, changes to the current schedule and cost. Budai et al. (2010) minimizes “off-balances” of rolling stock units, i.e., deviations from the target inventory level of a certain rolling stock type at a certain station at the end of the rescheduling period. Tomiyama et al. (2012a) solves rolling stock rescheduling by a combination of constraint programming and heuristics. Nielsen et al. (2012) proposes a rolling time horizon approach to rolling stock rescheduling. The approach can be viewed as online optimization (refer to Borodin and El-Yaniv (2005) on online optimization). Kroon et al. (2015) takes passenger’s behavior that depends on the seat capacity of each train into consideration, and proposes iterative heuristics which assigns a sufficient number of rolling stock units and estimates passenger’s behavior. The two rolling stock rescheduling algorithms are based on the scheduling algorithm by Fioole et al. (2006). Coupling and uncoupling of units are taken into account in these approaches, whereas routing inside depots and stations is omitted. Inspection or maintenance is not mentioned, either.

Maróti and Kroon (2005) discusses assignment of rolling stock units to a timetable in a situation where some of the units require inspection or maintenance in the forthcoming one to three days and hence they have to be routed to a maintenance facility. The problem is called “maintenance routing.” It is modeled as an integer programming formulation of an integer multicommodity network flow problem and is solved by using an optimization solver. Maróti and Kroon (2007) presents another model of the maintenance routing, in which the concept of an admissible interchange of sequences of tasks of two or more rolling stock units is introduced. A set of admissible interchanges is given, and independent interchanges are selected so that the units which require maintenance should be routed to a maintenance facility. A heuristic algorithm for the problem is proposed. Difficulty of shunting is taken into account via cost of a flow in objective function only in the former model, whereas an interchange corresponds to a group of shunting operations at a station in the latter model. The two models assume that an inspection is carried out at most once for a unit.

Yasumoto et al. (2009) presents a field trial of a rolling stock scheduling and rescheduling system. The algorithm is based on the metaheuristics presented by Otsuki et al. (2010). No computational results are described. Coupling and uncoupling is not taken into account. Tomiyama et al. (2012b) constructs a network whose vertex is a train task and a path is a sequence of tasks of a unit, and finds a solution by modifying the paths heuristically. The numerical results indicates that the inspection constraint is not satisfied for all the units. Shunting is not taken into account. In these two models, one train task is a schedule of a rolling stock unit for about one day, and new schedules for ten to 30 days are proposed. One or more inspections are proposed in the rescheduled sequence of tasks.

Aircraft rescheduling has been vigorously discussed. Clausen et al. (2010) provides an extensive review of disruption management in the industry. Due to a small numbers of available aircrafts, most of the aircraft rescheduling studies include timetable rescheduling. In some models, the aircraft circulation part is based on an integer programming formulation of the set partitioning which is solved by column generation, or an integer multicommodity network flow problem. Clausen (2007) points out the similarity of the disruption management process as well as rolling stock rescheduling models and algorithms between the railway and airline industry. On the other hand, it also indicates the difference in the problem scale, the computation time limit and above all the dimension of the infrastructure. Rolling stock runs on a one-dimensional railway. An overview of disruption management in the maritime industry is presented by Qi (2015). Brouer et al. (2013) and Li et al. (2015) study vessel rescheduling, which include timetable rescheduling. General vehicle rescheduling on a road network is reviewed by Visentini et al. (2014). Huisman and Wagelmans (2006) presents dynamic and simultaneous scheduling of vehicles and crews. It should be noted that, in general, vehicles can move between any two points on a road network, which does not hold true on a railway networks.

In our rolling stock rescheduling, the fixed updated timetable is given, since the simultaneous optimization is intractable in practical cases in Japan or the timetable rescheduling may be done a different entity. We distinguish coupling and uncoupling of units as well as routing inside depots and stations from the rolling stock rescheduling, and discuss it in shunting rescheduling. In the railway industry, our rolling stock rescheduling is the first approach that formulates the rescheduling as an integer programming problem which can be considered as a variant of the set partitioning problem. We apply column generation and the set-covering relaxation to solve the problem in acceptable time. The inspection considered in our rolling stock rescheduling is based on regulations in Japan, which is mentioned by Planning Systems Laboratory, Railway Technical Research Institute (2005), and is almost the same as the daily inspection (DI, level 1) discussed in rolling stock scheduling in Taiwan by Lai et al. (2015). The DI is required after any unit is operated for three days, and two or more DIs are considered in the scheduling. In our rescheduling, we have to let the rescheduling period be tens of hours when we consider night trains or freight trains whose running time amounts to the same magnitude. The rescheduled sequence of train tasks of a rolling stock unit is connected to the current sequence of tasks after the rescheduling period ends and the connected sequence of tasks has to satisfy the inspection constraint. These may require two or more inspections in the rescheduling period. In the column generation of our algorithm, a sequence of tasks which satisfies the inspection constraint is generated in polynomial time, by carrying out the necessary number of inspections at a proper interval. Recall that our algorithm does not necessarily provide an optimal solution for every instance due to a limited number of columns to be generated. It provides an optimal solution or decides the infeasibility of the instance correctly, however, when the column generation terminates with an integral solution.

4.3 Problem Description

4.3.1 Rescheduling Network

We construct the rolling stock rescheduling network $\mathcal{N} := (V, E, c, f, H)$ from the rescheduling situation on a railway line or a railway network on which we focus on. The notation is summarized in Tables 4.1 and 4.2. Then rescheduling start time is h_0 and the rescheduling period is h. The set of vertices V is defined as $V := K \cup I \cup \widehat{I} \cup S \cup D$. Let K be the set of rolling stock units involved in the rescheduling. For each $k \in K$, the time ready for the rescheduling is $\texttt{Ready\_time}(k)$, and the station where k is at $\texttt{Ready\_time}(k)$ is $\texttt{Ready\_sta}(k)$. It holds that $h_0 \leq \texttt{Ready\_time}(k) \leq h_0 + h$, otherwise we exclude k from K. We let $\texttt{PrevIns\_time}(k)$ be the time when the previous inspection of k is carried out and $\texttt{Ins\_int}(k)$ the maximum interval allowed between two consecutive inspections. The latter ordinarily depends on the type of a rolling stock unit. We let I be the set of trains tasks and $\widehat{I}$ be deadhead train tasks to be rescheduled. It consists of the train tasks in the current sequence of tasks of each $k \in K$ which have not yet been operated at $\texttt{Ready\_time}(k)$ or canceled as well as train tasks to which any rolling stock unit has not yet been assigned (they may be divided into several tasks at certain stations) if $h_0 < \texttt{Arr\_time}(i) < h_0 + h$ holds, where $\texttt{Arr\_time}(i)$ is the arrival time of train task i. An element i of I or $\widehat{I}$ also has its departure time $\texttt{Dep\_time}(i)$, its departure station $\texttt{Dep\_sta}(i)$ and its arrival station $\texttt{Arr\_sta}(i)$. Any deadhead task $\hat{i} \in \widehat{I}$ can be canceled, i.e., we do not have to assign a rolling stock unit. When we operate the train as an out-of-service train, there can is a case where a few rolling stock unit is coupled. The maximum number of rolling stock units that can be assigned to $\hat{i} \in \widehat{I}$ is denoted by $\overline{n_i}$. In Japan, $\overline{n_i} = 1$ for passenger rolling stock units $\underline{n_i} = 2$ for freight locomotives ordinarily. For each vertex $v \in K \cup I \cup \widehat{I}$, an inspection of the rolling stock units can be carried out after v when $\texttt{Arr\_sta}(v)$ is a a rolling stock depot or a station has facilities for the inspection. For every such v, we create inspection vertex s. The vertex has $\texttt{Prev\_task}(s) := v$ and has station $\texttt{Sta}(s) := \texttt{Arr\_sta}(v)$. When an inspection is carried out after $\texttt{Prev\_task}(s)$ in the current schedule of some rolling stock unit, we can regard the corresponding s as a scheduled inspection. Otherwise, it is an unscheduled inspection. The vertex also has the completion time of the inspection $\texttt{FinishIns\_time}(s)$, which is the arrival time of v plus the minimum time required for the inspection. The set of the inspection vertices are denoted by S. Since each rolling stock unit has its current sequence of tasks, its first task after the rescheduling period expires, or the task whose departure time is earlier than the completion time and arrival time is later than the completion time respectively, is definable. For a reserve rolling stock unit, the task is to be in its home depot. We call such a task the convergence task and the set of the convergence tasks are denote by D. It holds that $|D| = |K|$ by the definition. Each convergence task $d \in D$ has rolling stock unit which is planned to be assigned $\texttt{Current\_unit}(d)$ in the current rolling stock schedule and next inspection time $\texttt{NextIns\_time}(d)$ $(\geq h_0 + h)$ when a rolling stock unit is operated in the current sequence of tasks following d. The vertex has also $\texttt{Dep\_time}(d)$ and $\texttt{Dep\_sta}(d)$. The preparation time $\texttt{Prep\_time}(i_1, i_2)$ are required between pairs of train tasks (including deadhead and convergence tasks) $(i_1, i_2) \in (I \cup \widehat{I}) \times (I \cup \widehat{I} \cup D)$.

We next define the directed edge set E. The first type of the edges is drawn from a rolling stock unit to a train task. For $(k, i) \in K \times (I \cup \widehat{I} \cup D)$, the pair is an element of E if the following condition is satisfied:

$$\texttt{Ready\_sta}(k) = \texttt{Dep\_sta}(i) \text{ and } \texttt{Ready\_time}(k) \leq \texttt{Dep\_time}(i).$$

Let us consider pair of train task vertex pair including deadhead tasks and convergence tasks $(i_1, i_2) \in (I \cup \widehat{I}) \times (I \cup \widehat{I} \cup D)$. We let $(i_1, i_2) \in E$ if

$$\begin{aligned} \texttt{Arr\_sta}(i_1) &= \texttt{Dep\_sta}(i_2) \\ \text{and } \texttt{Arr\_time}(i_1) + \texttt{Prep\_time}(i_1, i_2) &\leq \texttt{Dep\_time}(i_2) \end{aligned}$$

holds. For each inspection vertex s, we draw $(\texttt{Prev\_task}(s), s) \in (K \cup I \cup \widehat{I}) \times S$, which means an inspection after the end of its previous task. Edge $(s, i) \in S \times (I \times \widehat{I} \cup D)$ is a member of E if it satisfies the condition:

$$\texttt{Sta}(s) = \texttt{Dep\_sta}(i) \text{ and } \texttt{FinishIns\_time}(s) \leq \texttt{Dep\_time}(i).$$

Note that there are no edge which starts at a convergence vertex.

Figure 4.1 is an example of a time-space diagram of the current timetable of Trains 1M–6M and 8M and sequences of rolling stock Units 1 and 2 on a railway line. Unit 1 is at Station B and Unit 2 is at Station C at the current time. We assume that an

Table 4.1 Notation for rolling stock rescheduling network (1 of 2)

$\mathbb{R}_+$	set of nonnegative real numbers
h_0	rescheduling start time
h	rescheduling period
$\mathcal{N} := (V, E, c, f, H)$	rolling stock rescheduling network
$V := K \cup I \cup \widehat{I} \cup S \cup D$	set of vertices
K	set of rolling stock units
`Ready_time(`k`)`	time ready for rescheduling for $k \in K$
`Ready_sta(`k`)`	depot or station where $k \in K$ is at `Ready_time(`k`)`
`PrevIns_time(`k`)`	previous inspection time of $k \in K$
`Ins_int(`k`)`	max. interval allowed between two consecutive inspections of $k \in K$
I	set of train tasks
$\widehat{I}$	set of deadhead train tasks
`Dep_time(`i`)`	departure time of $i \in I \cup \widehat{I}$
`Dep_sta(`i`)`	departure station or depot of $i \in I \cup \widehat{I}$
`Arr_time(`i`)`	arrival time of $i \in I \cup \widehat{I}$
`Arr_sta(`i`)`	departure station or depot of $i \in I \cup \widehat{I}$
$\overline{n_{\hat{i}}}$	max. number of rolling stock units that can be assigned to $\hat{i} \in \widehat{I}$
S	set of inspection vertices
`Prev_task(`s`)`	previous rolling stock unit or train task of $s \in S$
`Sta(`s`)`	depot or station at which $s \in S$ is carried out
`FinishIns_time(`s`)`	time at which $s \in S$ is to be finished
D	set of convergence train tasks
`Dep_time(`d`)`	arrival time of $d \in D$
`Dep_sta(`d`)`	arrival station or depot of $d \in D$
`Current_unit(`d`)`	rolling stock unit which is planned to be assigned to $d \in D$
`NextIns_time(`d`)`	next inspection time in sequence of tasks following $d \in D$
`Prep_time(`i_1, i_2`)`	preparation time required between $(i_1, i_2) \in (I \cup \widehat{I}) \times (I \cup \widehat{I} \cup D)$
E	set of directed edges

inspection can be carried out at Station A and Station C, and the label "Ins." indicates a scheduled inspection. Suppose here that the departure of train 2M at Station C is updated and delayed for minutes, to wait for passengers from another railway line for instance. Figure 4.2 shows this situation, which will cause Unit 2 to miss Train 4M unless the assigned unit is changed. In this case, no extra train delay will occur if Unit 1 is assigned to 4M and the following train tasks and Unit 2 is assigned to 1M and the following. Figure 4.3 is the rolling stock rescheduling network constructed from Figure 4.2. The

Table 4.2 Notation for rolling stock rescheduling network (2 of 2)

$c : K \times E \to \mathbb{R}_+$	rescheduling cost of $e \in E$ for $k \in K$
ω_1	positive cost value of assignment of different rolling stock unit at the beginning of duty
ω_2	positive cost value of assignment of different rolling stock unit in the middle of duty
ω_3	positive cost value of assignment of different type of rolling stock unit
ω_4	positive cost value of unscheduled inspection
ω_5	positive cost value of assignment of different rolling stock unit at convergence task
$f : I \cup D \to \mathbb{R}_+$	importance value of $i \in I \cup D$
$H^k : (\{k\} \cup S) \to 2^{(I \cup \widehat{I} \cup D)}$	inspection function
k-D path	path from $k \in K$ to any element of D on $\mathcal{N}$
$O_p^k(s_m, s_{m+1})$	set of vertices between inspection s_m and next inspection s_{m+1} in k-D path p
$O_p^k(s_n)$	set of vertices after final inspection s_n in k-D path p
P^k	set of feasible paths for $k \in K$ on $\mathcal{N}$
c_p^k	rescheduling cost of $p \in P^k$ for $k \in K$, defined as $c_p^k := \cup_{e \in p \cap E} c_e^k$
f_p^k	importance value of $p \in P^k$ for $k \in K$, defined as $f_p^k := \cup_{i \in p \cap (I \cup D)} f_v$
B	set of depots and stations at which inspection can be carried out
T	set of time spans
u_t^b	inspection capacity at $b \in B$ at $t \in T$

vertices of this network consist of the rolling stock units (k_1 and k_2), the tasks to be covered (from i_1 to i_6 and i_8), the scheduled inspection (s_2) after the arrival of Train 4 at Station A, unscheduled inspections (s_1 and s_3 as well as s_4), and the convergence tasks (d_5 and d_8). The directed edges are drawn between the vertices if the constraint on the time and the station is satisfied.

Let $\mathbb{R}_+$ be a set of nonnegative real numbers, and the cost function $c : K \times E \to \mathbb{R}_+$ returns a nonnegative value when a rolling stock unit and an edge are input. For simplicity, we substitute c_e^k for $c(k, e)$, and it means the cost for rolling stock unit r k to traverse edge e. We set $c_e^k := 0$ for edge $e \in E \cap ((K \cup I \cup \widehat{I}) \times (I \cup \widehat{I} \cup D))$ when its endpoint tasks coincide with a rolling stock unit and its next task or two consecutive train tasks in the current sequence of any rolling stock unit. Some positive cost is set otherwise, since it indicates a change of rolling stock assignment. We prepare two cost values ω_1 and ω_2 with $\omega_1 < \omega_2$. When edge e connects the last task of one duty with the first task of another duty, we select ω_1. Otherwise, we do ω_2. There are several types of rolling stock units

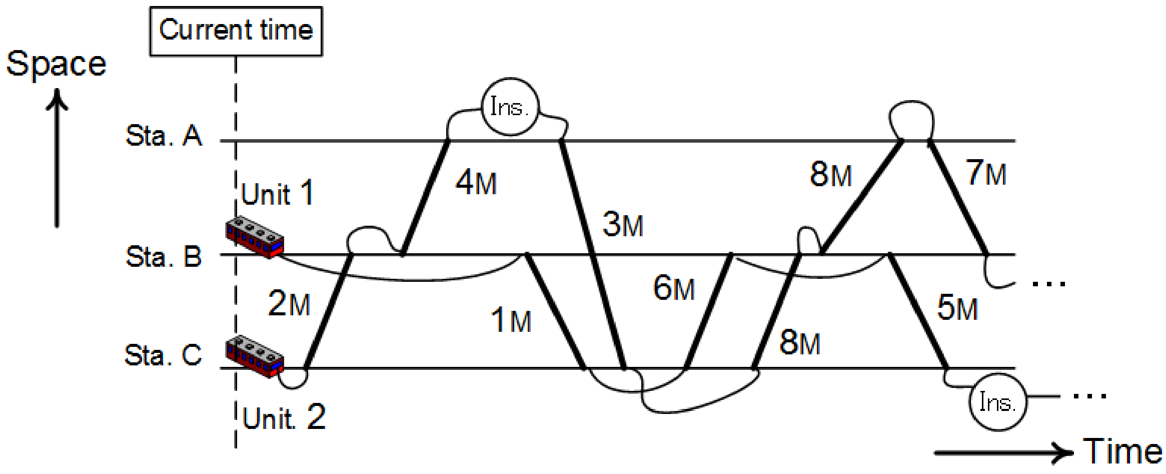

Fig. 4.1 Time-space diagram of current timetable and sequences of tasks of rolling stock units

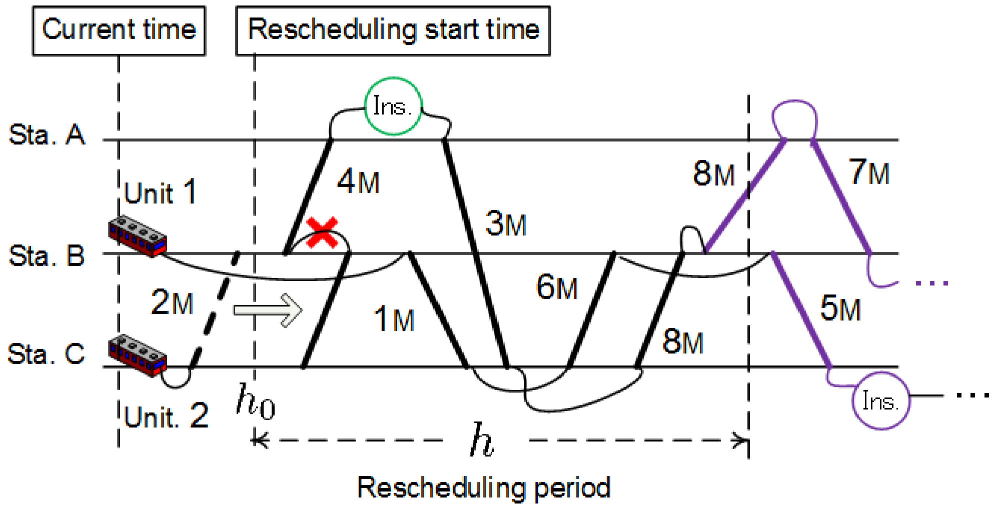

Fig. 4.2 Updated timetable

operated in the same line or the same network, and an additional cost ω_3 is imposed when a rolling stock unit is assigned to a task which is assigned to a different type of a rolling stock unit in the current sequence of tasks. For $(v, i) \in E \cap ((K \cup I \cup \widehat{I}) \times (I \cup \widehat{I} \cup D))$, if rolling stock unit k is not permitted to be assigned to i, we let $c_e^k := \infty$. For edge $(v, s) \in E \cap ((K \cup I \cup \widehat{I}) \times S)$ from a rolling stock unit or a task to an inspection vertex, its cost is zero when s is a scheduled inspection and is ω_3 otherwise. The cost of edge $(s, i) \in E \cap (S \times (I \cup \widehat{I} \cup D))$ from an inspection vertex to a task vertex is equal to that of $(\texttt{Prev\_task}(s), i)$. The cost of an edge ending at a convergence task, denoted by $c_{(v,d)}^k$ for $(v, d) \in E \cap ((K \cup I \cup \widehat{I} \cup S) \times D)$, depends on the rolling stock unit k. The assignment of a different rolling stock unit from the currently scheduled one can be permitted, though

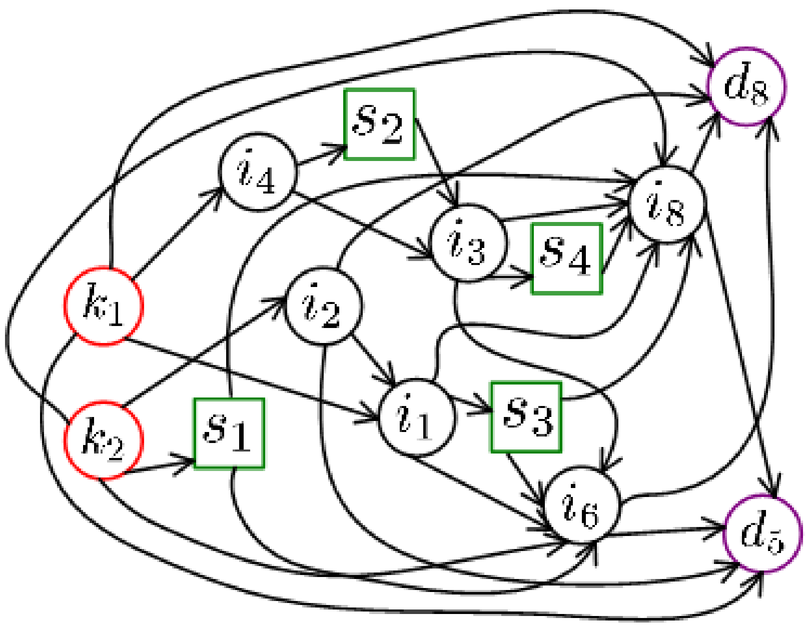

Fig. 4.3 Rolling stock rescheduling network

the assignment of the planned one is much preferred. Hence we set $c^k_{(v,d)} := 0$ when $\texttt{Current\_unit}(d) = k$ and $c^k_{(v,d)} := \omega_5$ otherwise. Let $f : I \cup \widehat{I} \to \mathbb{R}_+$ be importance value of $i \in I \cup D$. Note that the importance value of a deadhead train task is zero.

4.3.2 Feasible Paths and Inspection Capacity

The cost of path p of rolling stock unit $k \in K$ is denoted by c^k_p and it is defined as $c^k_p := \cup_{e \in p \cap E} c^k_e$. Then p with $c^k_p = 0$ is equivalent to the current sequence of tasks of k. A bigger value of c^k_p means that degree of the modification is large. Similarly, the importance of path p of rolling stock unit $k \in K$ is denoted by f^k_p and it is defined as $f^k_p := \cup_{i \in p \cap (I \cup D)} f_i$.

Recall here, that any rolling stock unit has to be inspected for every inspection interval $\texttt{Ins\_int}(k)$. To model the periodic inspection, we first prepare the inspection function $H^k : (\{k\} \cup S) \to 2^{(I \cup \widehat{I} \cup D)}$ such that, for each rolling stock unit $k \in K$ and each inspection $s \in S$,

$$\begin{aligned} H^k(k) := &\{i \in I \cup \widehat{I} \mid \texttt{PrevIns\_time}(k) + \texttt{Ins\_int}(k) \geq \texttt{Arr\_time}(i)\} \\ &\cup \{d \in D \mid \texttt{PrevIns\_time}(k) + \texttt{Ins\_int}(k) \geq \texttt{NextIns\_time}(d)\}, \\ H^k(s) := &\{i \in I \cup \widehat{I} \mid \texttt{FinishIns\_time}(s) + \texttt{Ins\_int}(k) \geq \texttt{Arr\_time}(i)\} \\ &\cup \{d \in D \mid \texttt{FinishIns\_time}(s) + \texttt{Ins\_int}(k) \geq \texttt{NextIns\_time}(d)\}. \end{aligned}$$

The set $H^k(k)$ is the set of tasks, without any inspection, to which rolling stock unit k

is permitted to be assigned to. The set $H^k(s)$ includes the tasks to which k is permitted to be assigned to after the inspection s is carried out. Note that $H^k(v_1) \subseteq H^k(v_2)$ is equivalent to $|H^k(v_1)| \leq |H^k(v_2)|$ for any $v_1, v_2 \in \{k\} \cup S$, since the function is based on time. We next define a path from $k \in K$ to any element of D on $\mathcal{N}$ as a *k-D path.* Let p be a k-D path, and we name the rolling stock and the inspection vertices contained in p $s_0, s_1, \ldots, s_n$ in order of their appearance where we $s_0 = k$. Let $O_p^k(s_m, s_{m+1}) \subseteq I \cup \widehat{I}$ be the tasks between s_m and s_{m+1} in p, and $O_p^k(s_n) \subseteq I \cup \widehat{I} \cup D$ the ones after the last inspection s_n. We then consider that path p is a *feasible path* if $c_p^k < \infty$ and p satisfies the following condition:

$$
\begin{aligned}
&H^k(s_m) \supseteq N^p(s_m, s_{m+1}) \quad \forall m \in \{0, \ldots, n-1\}, \\
&H^k(s_n) \supseteq N^p(s_n).
\end{aligned}
$$

The rolling stock unit is properly inspected in a feasible path. We let P^k as the set of feasible paths for $k \in K$.

The limited number of facilities and workers available for inspections restrict the number of inspections carried out at the same time. We let B be the set of rolling stock depots and the stations at which inspections can be carried out, and T be time span set. The capacity of inspections in depot or station $b \in B$ at time span $t \in T$ is denoted by u_t^b.

4.3.3 Problem Definitions

The rescheduling network $\mathcal{N}$ is given, and we define the rolling stock rescheduling as to assign each rolling stock unit $k \in K$ one feasible path $p \in P^k$, to exactly cover each of the train tasks $i \in I \cup D$ by one path and each of the deadhead train tasks $\hat{i} \in \widehat{I}$ by at most $\overline{n_i}$ paths, and not to carry out u_t^b inspections at any rolling stock depot or station $b \in B$ or at any time span $t \in T$, so that the sum of the cost of the assigned path c_p^k for all k is minimized. If it is not possible, we solve the rolling stock uncovered train tasks detection. It is defined as to maximize the sum of the importance of tasks f_p^k covered by p of all k, whose necessary conditions are that each k is assigned to one feasible path or no path, that each $i \in I \cup D$ is covered by at most one path and each $\hat{i} \in \widehat{I}$ be at most $\overline{n_i}$ paths, and that the inspection capacity is satisfied. The first condition is to make the problem feasible in a case where the set of feasible paths is empty for a certain rolling stock unit.

4.3.4 Complexity

We discuss here the complexity of the decision problem part of our rolling stock rescheduling problem. Maróti and Kroon (2005) shows that it is $\mathcal{NP}$-complete to decide if the vertices of a directed acyclic graph can exactly be covered by paths which satisfies the following conditions: each path starts at a source (a vertex with no incoming edges) and ends at a sink (a vertex with no outgoing edges), and one of its paths connects a given source with a given sink. In the rolling stock rescheduling problem, consider a case where $\widehat{I} = S = \varnothing$ holds and for every element of I there exist edges which start and end at the vertex respectively. Then the sources in the network are K and the sinks are D. We also assume that $H^{k_0}(k_0) = I \cup \{d_0\}$ for some $k_0 \in K$ and $d_0 \in D$, and that $H^k(k) = I \cup D$ for any other $k \in K \setminus \{k_0\}$. Then our problem is feasible if and only if the task vertices can exactly be covered by the paths which start at the sources and end at the sinks and one of its paths connects k_0 with d_0, which is equivalent to the $\mathcal{NP}$-complete case.

4.4 Integer Programming Formulations

4.4.1 Rolling Stock Rescheduling

We formulate the rolling stock rescheduling as an integer programming problem. The notation is summarized in Table 4.3. We let constant a^k_{ip} be

$$a^k_{ip} := \begin{cases} 1 & \text{if train task } i \in I \cup \widehat{I} \cup D \text{ is included in feasible path } p \in P^k \text{ of rolling stock} \\ & \text{unit } k \in K, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we define

$$a^{bk}_{tp} := \begin{cases} 1 & \text{if inspection at } b \in B \text{ at } t \in T \text{ is carried out in feasible path } p \in P^k \text{ of} \\ & \text{rolling stock } k \in K, \\ 0 & \text{otherwise.} \end{cases}$$

We then let x^k_p a decision variable defined as

$$x^k_p := \begin{cases} 1 & \text{if rolling stock unit } k \in K \text{ is assigned to feasible path } p \in P^k, \\ 0 & \text{otherwise.} \end{cases}$$

The notation is given, and we formulate the rolling stock rescheduling as an integer programming problem which can be considered as a variant of the set partitioning problem with another set-partitioning side constraint. The formulation is shown below and we call

Table 4.3 Notation for rolling stock rescheduling formulations and algorithm

Constants	
c_p^k	refer to Table 4.2
f_p^k	refer to Table 4.2
a_{ip}^k	1 if $i \in I \cup \widehat{I} \cup D$ is included in $p \in P^k$ for $k \in K$, 0 otherwise
a_{tp}^{bk}	1 if inspection at $b \in B$ at $t \in T$ is carried out in $p \in P^k$ for $k \in K$, 0 otherwise
$\overline{n_{\widehat{i}}}$	refer to Table 4.1
u_t^b	refer to Table 4.2
M	arbitrary large number
Decision Variables	
x_p^k	1 if $k \in K$ is assigned to $p \in P^k$, 0 otherwise
y	dummy variable
Temporary Sets and Values	
ℓ	iteration counter of column generation
P_ℓ^k	set of feasible paths for $k \in K$ on $\mathcal{N}$ at ℓ
I_ℓ	set of train tasks to which set-covering relaxation are applied at ℓ
$\widehat{I}_\ell$	set of deadhead train tasks to which set-covering relaxation are applied at ℓ
$\lambda_{i\ell}$	dual price of $i \in I \cup \widehat{I}$ at ℓ
μ_ℓ^k	dual price of $k \in K$ at ℓ
$\nu_{t\ell}^b$	dual price of $b \in B, t \in T$ at ℓ
γ	restoring parameter
Q_ℓ^k	set of feasible paths to be added for $k \in K$ on $\mathcal{N}$ at ℓ
r_ℓ^k	feasible shortest path length for $k \in K$ at ℓ
Formulations and Objective Values	
(RR^{IP})	rolling stock rescheduling problem
Z_{LB}	lower bound value of (RR^{IP})
(RD^{IP})	rolling stock uncovered train task detection problem
(RR_ℓ^{LP})	restricted master problem of (RR^{IP}) at ℓ
Z_ℓ^{LP}	objective value of (RR_ℓ^{LP}) at ℓ
(RSP_ℓ)	column generation subproblem of (RR^{IP}) at ℓ
(RR_ℓ^{IP})	restricted rolling stock rescheduling problem at ℓ
Z_ℓ^{IP}	objective value of (RR_ℓ^{IP})

it (RR^{IP}):

$$
(RR^{\mathsf{IP}}) \quad
\begin{array}{lllr}
\text{minimize} & \displaystyle\sum_{k\in K}\sum_{p\in P^k} c_p^k x_p^k & & (4.1)\\
\text{subject to} & \displaystyle\sum_{k\in K}\sum_{p\in P^k} a_{ip}^k x_p^k = 1 & \forall i \in I \cup D, & (4.2)\\
& \displaystyle\sum_{k\in K}\sum_{p\in P^k} a_{\hat{i}p}^k x_p^k \leq \overline{n_{\hat{i}}} & \forall \hat{i} \in \widehat{I}, & (4.3)\\
& \displaystyle\sum_{p\in P^k} x_p^k = 1 & \forall k \in K, & (4.4)\\
& \displaystyle\sum_{k\in K}\sum_{p\in P^k} a_{tp}^{bk} x_p^k \leq u_t^b & \forall b \in B \;\; \forall t \in T, & (4.5)\\
& x_p^k \in \{0,1\} & \forall k \in K \;\; \forall p \in P^k. & (4.6)
\end{array}
$$

The objective function (4.1) is the total cost of feasible paths to which the rolling stock units are assigned, and we try to minimize it. The constraint (4.2) means that all the train tasks have to be covered by exactly one feasible path of the rolling stock units. This is the main set-partitioning constraint. The deadhead train tasks can be covered by at most $\overline{n_{\hat{i}}}$ paths, and the constraint (4.3) indicates this. Each rolling stock unit is assigned to one feasible path in its feasible path set, which is stated in the constraint (4.4). It can be seen as set-partitioning of K. The constraint (4.5) indicates that the number of inspected rolling stock units at $b \in B$ at $t \in T$ does not exceed the inspection capacity u_t^b. The constraint (4.6) restrict the value of the decision variables to either zero or one. We let Z^{IP} be the objective value of (RR^{IP}).

4.4.2 Rolling Stock Uncovered Train Task Detection

We call the integer programming formulation of the rolling stock uncovered train task detection problem (RD^{IP}). It is shown below:

$$
(RD^{\mathsf{IP}}) \quad
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{k\in K}\sum_{p\in P^k} f_p x_p^k & \\
\text{subject to} & \displaystyle\sum_{k\in K}\sum_{p\in P^k} a_{ip}^k x_p^k \le 1 & \forall i \in I \cup D, \\
& \displaystyle\sum_{k\in K}\sum_{p\in P^k} a_{\hat{i}p}^k x_p^k \le \overline{n_{\hat{i}}} & \forall \hat{i} \in \widehat{I}, \\
& \displaystyle\sum_{p\in P^k} x_p^k \le 1 & \forall k \in K, \\
& \displaystyle\sum_{k\in K}\sum_{p\in P^k} a_{tp}^{bk} x_p^k \le u_t^b & \forall b \in B \;\; \forall t \in T, \\
& x_p^k \in \{0,1\} & \forall k \in K \;\; \forall p \in P^k.
\end{array}
$$

The constraint on $I \cup D$ is called the set-packing constraint. Note that this problem is always feasible since $x_p^k = 0$ for all $k \in K$ and $p \in P^k$ satisfies the constraints, provided that we define the sum of x_p^k is zero when $P^k := \varnothing$. Any train task $i \in I$ that is note covered by any path means that the task cannot be operated due to the lack of rolling stock units. The solution to (RD^{IP}) thus helps us decide which train should be delayed or canceled in timetable rescheduling.

4.5 Algorithm

4.5.1 Overall Algorithm and Initialization

We apply column generation to (RR^{IP}) since all the elements of the feasible path set P^k for each $k \in K$ are not known in advance, and it will take large amount of time to enumerate them. The overall algorithm is displayed in Figure 4.4.

At Step 1 of the algorithm, we let ℓ be an iteration counter of the column generation and set $\ell := 1$. We also let Z_{LB} be a lower bound value of (DR^{IP}) and set $Z_{\mathsf{LB}} := 0$. A subset of P^k for each $k \in K$ at ℓ is denoted by P_ℓ^k and we set $P_1^k := \varnothing$. Let $I_\ell, \widehat{I}_\ell$ be subsets of $I, \widehat{I}$, respectively, and set $I_1, \widehat{I}_1 := \varnothing$. We introduce nonnegative parameter γ called restoring parameter which is relevant to the algorithm at Step 4. Its typical value is from 0.0 to 5.0.

4.5.2 Solving Restricted Master Problem

Let M be arbitrary large number and y be a nonnegative dummy variable. We also let x_p^k be a nonnegative continuous variable. We define the restricted master problem (RR_ℓ^{LP}) of

Step 1: *Initialization.*
Set $\ell := 1, Z_{\mathsf{LB}} := 0, P_\ell^k := \varnothing\ \forall k \in K, I_\ell := \varnothing, \widehat{I}_\ell := \varnothing$.
Let γ positive value.
Step 2: *Solving Restricted Master Problem.*
Solve (RR_ℓ^{LP}).
Let Z_ℓ^{LP} be objective value, $\lambda_{i\ell}, \mu_\ell^k, \nu_{t\ell}^b$ be dual prices.
Step 3: *Solving Column Generation Subproblem.*
If $\ell = 1$, construct $\mathcal{N}^\star$.
Let $-\lambda_{i\ell}, -\mu_\ell^k, -\nu_{t\ell}^b$ be cost on corresponding vertices of $\mathcal{N}^\star$.
Parallel for $k \in K$,
Solve shortest path problem on $\mathcal{N}^\star$.
Let r_ℓ^k be feasible shortest path length.
Let Q_ℓ^k be set of feasible paths with negative cost.
Step 4: *Lower Bound Update and Restoring Constraints.*
Set $Z_{\mathsf{LB}} := \max\{Z_{\mathsf{LB}},\ Z_\ell^{\mathsf{LP}} + \sum_{k\in K} r_\ell^k\}$.
If $\gamma \times (Z_\ell^{\mathsf{LP}} - Z_{\mathsf{LB}}) \geq -\sum_{k\in K} r_\ell^k$, then
$I_{\ell+1} := I_\ell \cup$ (tasks in $(I \cup D) \setminus I_\ell$ violating (4.9) in solution to (RR_ℓ^{LP})).
$\widehat{I}_{\ell+1} := \widehat{I}_\ell \cup$ (tasks in $\widehat{I} \setminus \widehat{I}_\ell$ violating (4.11) in solution to (RR_ℓ^{LP})).
Else, $I_{\ell+1} := I_\ell, \widehat{I}_{\ell+1} := \widehat{I}_\ell$.
Step 5: *Termination of Column Generation.*
If ($Z_\ell^{\mathsf{LP}} > Z_{\mathsf{LB}}$ or $I_{\ell+1} \neq I_\ell$ or $\widehat{I}_{\ell+1} \neq \widehat{I}_\ell$), then
Set $P_{\ell+1}^k := P_\ell^k \cup Q_\ell^k\ \forall k \in K, \ell := \ell + 1$ and Go to Step 2.
Else if $Z_\ell^{\mathsf{LP}} = M$, then
Output "Rolling stock rescheduling infeasible." and Go to Step 7.
Step 6: *Solving Restricted Rolling Stock Rescheduling Problem.*
Solve (RR_ℓ^{IP}).
Let Z_ℓ^{IP} be objective value.
If $Z_\ell^{\mathsf{IP}} < M$, then
Output rolling stock rescheduling solution and Stop.
Else,
Output "Rolling stock rescheduling solution not found."
Step 7: *Solving Rolling Stock Uncovered Train Task Detection Problem.*
Solve (RD^{IP}) by column generation.
Output rolling stock uncovered train task detection solution and Stop.

Fig. 4.4 Rolling stock rescheduling algorithm

(RR^{IP}) as

$$
(RR_\ell^{\mathsf{LP}}) \quad
\begin{array}{lrll}
\text{minimize} & \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} c_p^k x_p^k + My & & (4.7) \\
\text{subject to} & \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} a_{ip}^k x_p^k + y \geq 1 & \forall i \in (I\cup D)\setminus I_\ell, & (4.8) \\
& \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} a_{i'p}^k x_p^k + y = 1 & \forall i' \in I_\ell, & (4.9) \\
& \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} a_{\hat{i}p}^k x_p^k + \overline{n_{\hat{i}}} y < \infty & \forall \hat{i} \in \widehat{I}\setminus \widehat{I}_\ell, & (4.10) \\
& \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} a_{\hat{i}'p}^k x_p^k + \overline{n_{\hat{i}}} y \leq \overline{n_{\hat{i}}} & \forall \hat{i}' \in \widehat{I}_\ell, & (4.11) \\
& \displaystyle\sum_{p\in P_\ell^k} x_p^k + y = 1 & \forall k \in K, & (4.12) \\
& \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} a_{tp}^{bk} x_p^k + u_t^b y \leq u_t^b & \forall b \in B \;\; \forall t \in T, & (4.13) \\
& y \geq 0, \quad x_p^k \geq 0 & \forall k \in K \;\; \forall p \in P_\ell^k. & (4.14)
\end{array}
$$

This is a linear programming problem (refer to Hillier and Lieberman (2014), Nemhauser and Wolsey (1999) and Williams (2013) on linear programming), and is always feasible if we let $y = 1$ and $x_p^k = 0$ for all $k \in K, p \in P^k$. We relax part of the original set-partitioning constraint (4.2) and make it the set-covering constraint (4.8), since large problem instances of the set partitioning problem are known to be difficult to solve, according to Lübbecke and Desrosiers (2005). We also let that the tasks in $\widehat{I}$ be covered by more than $\overline{n_{\hat{i}}}$ rolling stock units. We call it the set-covering relaxation.

At Step 2, we solve (RR_ℓ^{LP}) by an optimization solver. We let Z_ℓ^{LP} be the objective value of (RR_ℓ^{LP}), $\lambda_{i\ell}$ be the dual price corresponding to train task i or $\hat{i}$ of the constraint set (4.8)–(4.11), μ_ℓ^k be that corresponding to the constraint set (4.12) and $\nu_{t\ell}^b$ be that corresponding to the constraint set (4.13).

4.5.3 Solving Column Generation Subproblem

After (RR_ℓ^{LP}) is solved to optimality, we decide whether the objective value Z_ℓ^{LP} can further be improved by adding new feasible paths to the set P_ℓ^k for any rolling stock unit at Step 3. We search for such a feasible path by modeling the column generation subproblem (RSP_ℓ) shown below, which is derived from the dual of (RR_ℓ^{LP}) and the linear relaxation of (RR^{IP}):

$$(RSP_\ell) \quad \begin{array}{|ll} \text{find} & k \in K \;\; p \in P^k \\ \text{such that} & c_p^k - \displaystyle\sum_{i \in I \cup \widehat{I} \cup D} a_{ip}^k \lambda_{i\ell} - \mu_\ell^k - \sum_{b \in B} \sum_{t \in T} a_{tp}^{bk} \nu_{t\ell}^b < 0. \end{array}$$

The important requirement for $p \in P^k$, i.e., the path is a member of the feasible path set, is that the inspection constraint has to be satisfied. We reduce the search problem to an unconstrained shortest path problem by extending the rolling stock rescheduling network $\mathcal{N}$ and introducing $\mathcal{N}^\star$ named the *extended network*. We construct $\mathcal{N}^\star$, at the first iteration of the rescheduling algorithm, as follows:

Step A: Sort $I \cup \widehat{I} \cup D$ in ascending order of `Arr_time(`i`)` (for $i \in I \cup \widehat{I}$) and `NextIns_time(`d`)` (for $d \in D$). For each i, let its ranking number on index `Ind(`i`)`.

Step B: For each $k \in K$, make $(|I \cup \widehat{I} \cup D| - |H^k(k)| + 1)$ copies of network $\mathcal{N}$ and call them $\mathcal{N}^{|H^k(k)|}, \ldots, \mathcal{N}^{|I \cup \widehat{I} \cup D|}$. Add superscript $j \in \{|H^k(k)|, \ldots, |I \cup \widehat{I} \cup D|\}$ to all the sets and the functions, the elements of the sets and the attributes of network $\mathcal{N}^j$.

Step C: For each k and j, let the endpoint i^j of every edge $(s^j, i^j) \in E^j \cap (S^j \times (I^j \cup \widehat{I}^j \cup D^j))$ of $\mathcal{N}^j$ be $i^{|H^k(s^j)|}$ of $\mathcal{N}^{|H^k(s^j)|}$.

Step D: For each j, delete every edge $(v^j, i^j) \in E^j \cap (V^j \times (I^j \cup \widehat{I}^j \cup D^j))$ of $\mathcal{N}^j$ if it satisfies `Ind(`i^j`)` $> j$.

An example of network $\mathcal{N}$ with one rolling stock unit is displayed in Figure 4.5 as an initial network. In this figure, we can assign the rolling stock unit k to the three train tasks i_1, i_2 and i_3 without carrying out any inspection. We assume here that the sum of `FinishIns_time(`s_1`)` of the inspection vertices s_1 and the inspection interval `Ins_int(`k`)` exceeds `Arr_time(`i_5`)` and not `NextIns_time(`d_6`)`, and that the sum of `FinishIns_time(`s_2`)` and `Ins_int(`k`)` exceeds `NextIns_time(`d_6`)`. The corresponding extended network $\mathcal{N}^\star$ is Figure 4.6. Path $k \to i_1 \to i_4 \to d_6$ or path $k \to i_2 \to i_5 \to d_6$ in Figure 4.5 is infeasible since it violates the inspection constraint at i_4 or i_5. Path $k \to i_1 \to s_1 \to i_4 \to d_6$ is not feasible either at some point of time in the sequence of train tasks in d_6. In this case, the only feasible path is $k \to i_2 \to s_2 \to i_5 \to d_6$ and the only path from the rolling stock unit vertex k^3 to a convergence task of $\mathcal{N}^\star$ is $k^3 \to i_2^3 \to s_2^3 \to i_5^6 \to d_6^6$ in Figure 4.6.

We next discuss a more general case. We define a set of path from $k^{|H^k(k)|}$ for $k \in K$ and to a member of $\cup_{j \in \{|H^k(k)|, \ldots, |I \cup \widehat{I} \cup D|\}} D^j$ on $\mathcal{N}^\star$ as *$k^\star$-$D^\star$ path*. We transform the

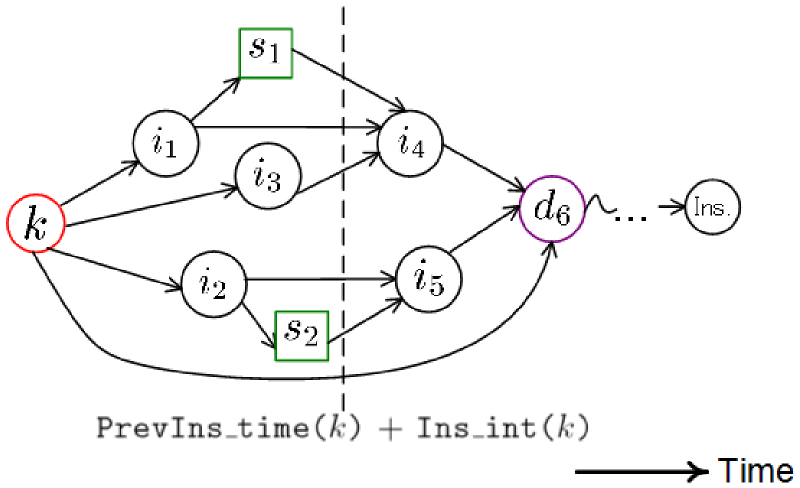

Fig. 4.5 Rolling stock rescheduling network with one unit $\mathcal{N}$

sequence of vertices of a $k^\star$-$D^\star$ path on $\mathcal{N}^\star$ into one on $\mathcal{N}$ in the following manner:

$$v^j \rightharpoonup v,$$

i.e., we simply delete the superscript of each vertex. This sequence forms a k-D path on $\mathcal{N}$ since any edge on $\mathcal{N}^\star$ is derived from $\mathcal{N}$. We transform the sequence of vertices of $p \in P^k$ on $\mathcal{N}$ into one on $\mathcal{N}^\star$ in the following manner:

$$\begin{aligned} k &\rightharpoonup k^{|H^k(k)|}, & i \in O^k_p(s_m, s_{m+1}) &\rightharpoonup i^{|H^k(s_m)|}, \\ s_{m+1} &\rightharpoonup i^{|H^k(s_m)|}, & i \in O^k_p(s_n) &\rightharpoonup i^{|H^k(s_n)|}. \end{aligned}$$

Then the following holds.

Proposition 4.1. *For all $k \in K$, every $k^\star$-$D^\star$ path on $\mathcal{N}^\star$ corresponds to a path in P^k on $\mathcal{N}$ and vice versa.*

Proof. A $k^\star$-$D^\star$ path p on $\mathcal{N}^\star$ for any k is given, and we first show that the path is included in P^k. For any vertex $i \in O^k_p(s_m, s_{m+1})$ between two inspections s_m and s_{m+1} in p, $\texttt{Ind}(i) \le |H^k(s_m)|$ holds from Steps C and D of the procedure of constructing $\mathcal{N}^\star$. Task i' with $\texttt{Ind}(i') = |H^k(s_m)|$ is a member of $H^k(s_m)$ (otherwise $H^k(s_m)$ has at most $\texttt{Ind}(i') - 1$ tasks whose $\texttt{Arr\_time}(*)$ is less than $\texttt{Arr\_time}(i)$ from the definition of the function H^k, a contradiction), and therefore $i \in H^k(s_m)$ holds from Step A. The same holds true for $i \in O^k_p(s_n)$, and p is shown to be a member of P^k.

The sequence of vertices in a feasible path $p \in P^k$ for any $k \in K$ are given, and we transform it into one on $\mathcal{N}^\star$. For any $i \in O^k_p(s_m, s_{m+1})$, it holds that $i \in H^k(s_m)$

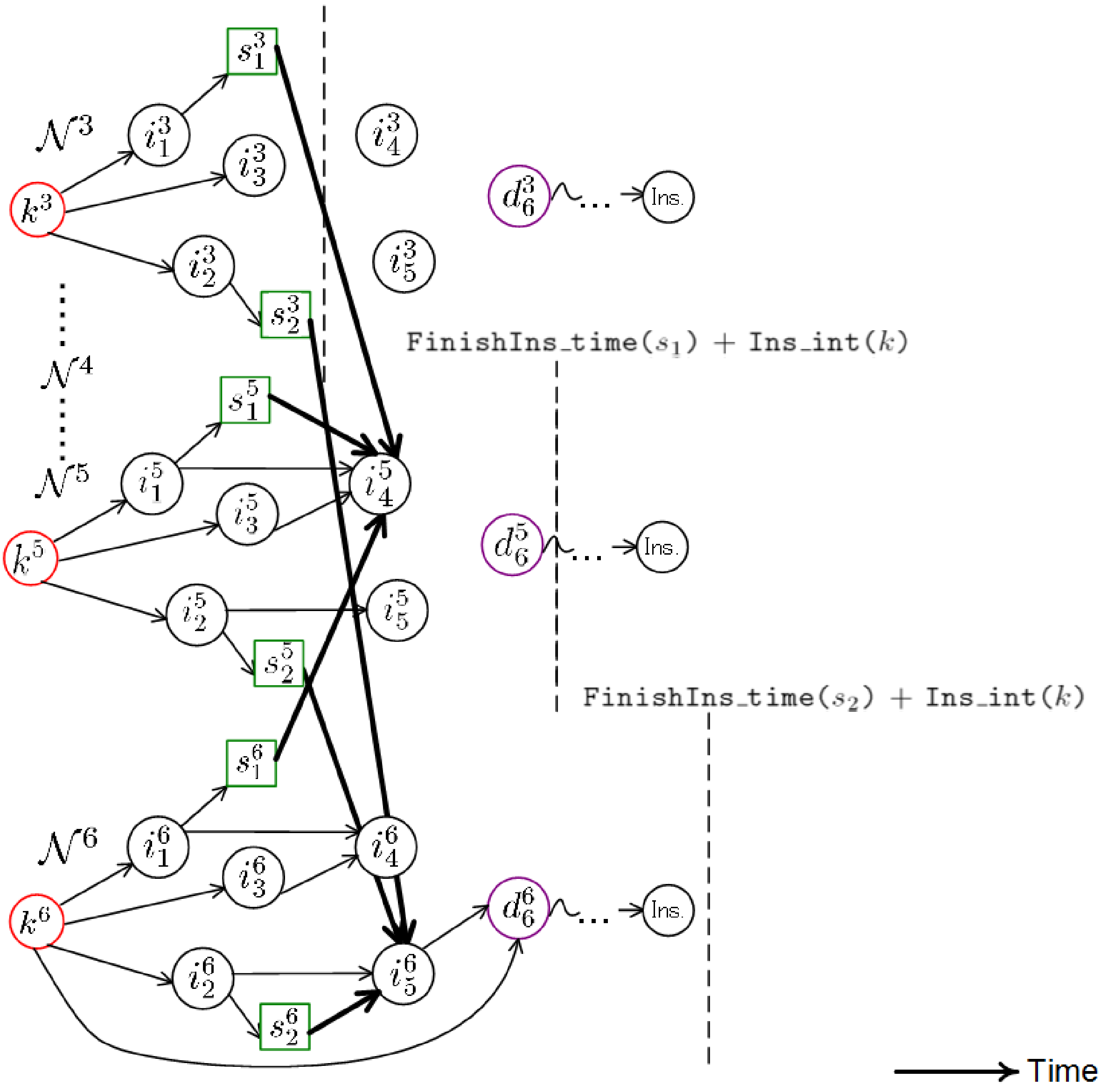

Fig. 4.6 Extended network $\mathcal{N}^{\star}$

by the definition of the inspection function, and that $\texttt{Ind}(i) \leq |H^k(s_m)|$ (otherwise $\texttt{Arr\_time}(i) > \texttt{Arr\_time}(i')$ with i' with $\texttt{Ind}(i') = |H^k(s_m)|$ and $i \notin H^k(s_m)$). The copied vertex $i^{|H^k(s_m)|}$ at Step B also has an index $\texttt{Ind}(i^{|H^k(s_m)|}) \leq |H^k(s_m)|$, and therefore the edges which end at $i^{|H^k(s_m)|}$ are not deleted from the partial network $\mathcal{N}^{|H^k(s_m)|}$ of $\mathcal{N}^{\star}$ at Step D. The same holds true for $i \in O_p^k(s_n)$. The edges which end at s_{m+1} are not deleted either. The vertices on $\mathcal{N}^{\star}$ which corresponds to the inspection vertex s_{m+1} and its next vertex in p are connected by the edge drawn at Step C and the edge connects between $\mathcal{N}^{|H^k(s_m)|}$ and $\mathcal{N}^{|H^k(s_{m+1})|}$. Therefore, the transformed sequence of vertices forms a $k^{\star}$-$D^{\star}$ path on $\mathcal{N}^{\star}$. □

Recall here that c_p^k in (RSP_ℓ), the cost of feasible path p of rolling stock unit $k \in K$, is defined as the sum of the cost of the edges included in p. We let $-\lambda_{i\ell}, -\mu_\ell^k$ be the cost of the corresponding vertices of $\mathcal{N}^\star$. We also let $-\nu_t^b$ be on inspection vertex s in the following manner: $b = \texttt{Sta}(s)$, and t is the time span which has the minimum value of $-\nu_t^b$ satisfying $t \subseteq [\texttt{Arr\_time(Prev\_task(}s\texttt{))},\texttt{Dep\_time(}v\texttt{)}]$ where v is a vertex such that $(s, v) \in E$. We can thus reduce (RSP_ℓ) to the shortest path problem from $k^{|H^k(k)|}$ to elements of D^j where $j \in \{|H^k(k)|, \ldots, |I \cup \widehat{I} \cup D|\}$ on the extended network $\mathcal{N}^\star$. Note that inspections can be carried out twice or more on our extended network, which is required when the rescheduling period is long. Since the shortest path of one rolling stock unit is independent of that of another, we solve the shortest path problem concurrently for several of K.

For an acyclic graph with α vertices and β edges, we can solve the shortest path problem in $O(\alpha + \beta)$ by Dijkstra's algorithm. Refer to Hillier and Lieberman (2014) as well as Nemhauser and Wolsey (1999) on the shortest path problem as well as Dijkstra's algorithm and Wolsey (1998) on those on a directed acyclic graph. For each $k \in K$, the order of vertices to be visited is, the rolling stock unit vertex $k^{|H^k(k)|}$, the train tasks in ascending order of $\texttt{Arr\_time(}i^j\texttt{)}$, $\texttt{FinishIns\_time(}s^j\texttt{)}$ and $\texttt{NextIns\_time(}d^j\texttt{)}$ where $j \in \{|H^k(k)|, \ldots, |I \cup \widehat{I} \cup D|\}$. The extended network $\mathcal{N}^\star$ has at most $|V|(|I \cup \widehat{I} \cup D| - |H^k(k)| + 1)$ vertices and $|E|(|I \cup \widehat{I} \cup D| - |H^k(k)| + 1)$ edges. Since $|K| = |D|$, $|S| \leq 1 + |I| + |\widehat{I}|$ by the construction rule of the vertices and $H^k(v)$ for $v \in \{k\} \cup S$ takes at most $1 + |S|$ different values among the $(|I \cup \widehat{I} \cup D| - |H^k(k)| + 1)$ candidates, the shortest path can be found in $O((1 + |I| + |\widehat{I}| + |D| + |S| + |E|)(|S| + 1) = O((|K| + |I| + |\widehat{I}| + |E|)|S|)$. In a special case where $h \leq \texttt{Ins\_int(}k\texttt{)}$ holds, i.e., the rescheduling period is not longer than the inspection interval of rolling stock unit k, we can solve the shortest path problem in $O((|K| + |I| + |\widehat{I}| + |E|)|K|)$ for each k. That is since $H^k(s) \geq |I \cup \widehat{I}|$ holds for any $s \in S$ and there is no edge from one vertex of $\mathcal{N}^{|H^k(k)|}$ to another vertex of $\mathcal{N}^j$ for any $j \in \{(|H^k(k)| + 1), \ldots, (|I \cup \widehat{I}| - 1)\}$.

Let r_ℓ^k be feasible shortest path length for $k \in K$ (we set $r_\ell^k := 0$ when $P^k = \varnothing$). We add feasible paths with negative cost to set Q_ℓ^k if they exist for some k. The feasible paths other than the shortest one which are obtained by Dijkstra's algorithm can also be added.

4.5.4 Lower Bound Update and Restoring Constraints

The shortest feasible paths found at ℓ is equal to the optimal solution to the Lagrangian relaxation problem of (RR^{IP}) with the constraints (4.2), (4.3), (4.5) being relaxed, if we regard the dual prices $\lambda_{i\ell}, \nu_{t\ell}^b$ as Lagrangian multipliers (refer to Desrosiers and Lübbecke (2005), Nemhauser and Wolsey (1999) and Wolsey (1998) on Lagrangian relaxation and duality). The formulation is as follows:

$$
\left|\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{k\in K}\sum_{p\in P^k} c_p^k x_p^k + \sum_{i\in I\cup D}\lambda_{i\ell}(1-\sum_{k\in K}\sum_{p\in P^k} a_{ip}^k x_p^k) & \\
& \displaystyle\qquad + \sum_{\hat{i}\in\hat{I}}\lambda_{\hat{i}\ell}(\overline{n_{\hat{i}}} - \sum_{k\in K}\sum_{p\in P^k} a_{ip}^k x_p^k) & \\
& \displaystyle\qquad + \sum_{b\in B\ \forall t\in T}\nu_{t\ell}^b(u_t^b - \sum_{k\in K}\sum_{p\in P^k} a_{tp}^{bk} x_p^k) & \\
\text{subject to} & \displaystyle\sum_{p\in P^k} x_p^k = 1 & \forall k\in K, \\
& x_p^k \in \{0,1\} & \forall k\in K\ \ \forall p\in P^k.
\end{array}\right.
$$

Let $\widetilde{x}_p^k$ be one if it is the shortest feasible path p of the rolling stock k and be zero otherwise, which is equivalent to the optimal solution to the Lagrangian problem. By the duality theorem, $r_\ell^k \leq 0$ for every k and the following holds:

$$
\begin{aligned}
& \sum_{k\in K}\sum_{p\in P^k} c_p^k\widetilde{x}_p^k + \sum_{i\in I\cup D}\lambda_{i\ell}(1-\sum_{k\in K}\sum_{p\in P^k} a_{ip}^k\widetilde{x}_p^k) + \sum_{\hat{i}\in\hat{I}}\lambda_{\hat{i}\ell}(\overline{n_{\hat{i}}} - \sum_{k\in K}\sum_{p\in P^k} a_{\hat{i}p}^k\widetilde{x}_p^k) \\
& + \sum_{b\in B\ \forall t\in T}\nu_{t\ell}^b(u_t^b - \sum_{k\in K}\sum_{p\in P^k} a_{tp}^{bk}\widetilde{x}_p^k) \\
= & \sum_{k\in K}\sum_{p\in P^k} c_p^k\widetilde{x}_p^k + \sum_{i\in I\cup D}\lambda_{i\ell} + \sum_{\hat{i}\in\hat{I}}\overline{n_{\hat{i}}}\lambda_{\hat{i}\ell} - \sum_{i\in I\cup\hat{I}\cup D}\sum_{k\in K}\sum_{p\in P^k} a_{ip}^k\lambda_{i\ell}\widetilde{x}_p^k \\
& + \sum_{k\in K}\mu_\ell^k - \sum_{k\in K}\mu_\ell^k\sum_{p\in P^k}\widetilde{x}_p^k + \sum_{b\in B\ \forall t\in T}\nu_{t\ell}^b u_t^b - \sum_{b\in B\ \forall t\in T}\sum_{k\in K}\sum_{p\in P^k} a_{tp}^{bk}\nu_{t\ell}^b\widetilde{x}_p^k \\
= & \sum_{i\in I\cup D}\lambda_{i\ell} + \sum_{\hat{i}\in\hat{I}}\overline{n_{\hat{i}}}\lambda_{\hat{i}\ell} + \sum_{k\in K}\mu_\ell^k + \sum_{b\in B\ \forall t\in T}\nu_{t\ell}^b u_t^b \\
& + \sum_{k\in K}\sum_{p\in P^k}(c_p^k - \sum_{i\in I\cup\hat{I}\cup D} a_{ip}^k\lambda_{i\ell} - \mu_\ell^k - \sum_{b\in B\ \forall t\in T} a_{tp}^{bk}\nu_{t\ell}^b)\widetilde{x}_p^k \\
= & Z_\ell^{\mathsf{LP}} + \sum_{k\in K} r_\ell^k.
\end{aligned}
$$

That is, the optimal value of the Lagrangian relaxation problem is equal to the objective value of (RR_ℓ^{LP}) plus the sum of the shortest path lengths. Since this value is a lower bound of (RR^{IP}), we can update Z_{LB} if $Z_{\mathsf{LB}} < Z_\ell^{\mathsf{LP}} + \sum_{k\in K} r_\ell^k$ at the beginning of Step 4.

We refer here to the value of restoring parameter γ and enlarge the task sets I_ℓ and $\widehat{I}_\ell$ which are relevant to the constraints (4.9) and (4.11) if the following inequality is satisfied:

$$\gamma \times (Z_\ell^{\mathsf{LP}} - Z_{\mathsf{LB}}) \geq -\sum_{k\in K} r_\ell^k.$$

Then we let the optimal solution to (RR_ℓ^{LP}) obtained at Step 2 be $\check{x}_p^k, \check{y}$, and update the task sets at the next restricted master problem ($RR_{\ell+1}^{\mathsf{LP}}$) in the following manner:

$$I_{\ell+1} := I_\ell \cup \{i \in I \cup D \mid \sum_{k \in K} \sum_{p \in P_\ell^k} a_{ip}^k \check{x}_p^k + \check{y} > 1\},$$
$$\widehat{I}_{\ell+1} := \widehat{I}_\ell \cup \{\hat{i} \in \widehat{I} \mid \sum_{k \in K} \sum_{p \in P_\ell^k} a_{\hat{i}p}^k \check{x}_p^k + \overline{n_{\hat{i}}}\check{y} > \overline{n_{\hat{i}}}\}.$$

When the inequality involving parameter γ is not satisfied, we let $I_{\ell+1} := I_\ell$ and $\widehat{I}_{\ell+1} := \widehat{I}_\ell$.

If Z_{LB} is updated at the begging of Step 4, $Z_{\mathsf{LB}} = Z_\ell^{\mathsf{LP}} + \sum_{k \in K} r_\ell^k$. The above inequality on γ can be rewritten as

$$\gamma \times (-\sum_{k \in K} r_\ell^k - \epsilon) \geq -\sum_{k \in K} r_\ell^k$$

where

$$\epsilon = \begin{cases} 0 & \text{if } Z_{\mathsf{LB}} \text{ is updated at } \ell, \\ Z_{\mathsf{LB}} - (Z_\ell^{\mathsf{LP}} + \sum_{k \in K} r_\ell^k) & \text{otherwise.} \end{cases}$$

Recall that $r_\ell^k \leq 0$ for every k and note that $\epsilon \geq 0$. For $0 \leq \gamma < 1$, the sets $I_{\ell+1}$ and $\widehat{I}_{\ell+1}$ are enlarged when $\sum_{k \in K} r_\ell^k = 0$ and consequently $\epsilon = 0$ since $Z_\ell^{\mathsf{LP}} \geq Z_{\mathsf{LB}}$. For $\gamma = 1$, the constraints are changed when $\sum_{k \in K} r_\ell^k = 0$ or the lower bound Z_{LB} is updated. If we set $\gamma > 1$, then the sets are also enlarged when the optimal value of the Lagrangian relaxation problem of (RR^{IP}) at ℓ is close to the current lower bound. From the viewpoint of the dual problem, the domain of the variable $\lambda_{i\ell} \geq 0$ for $i \in I \cup D$ is enlarged and becomes $\lambda_{i\ell} > -\infty$ by restoring the constraint for i. Hence our relaxation approach can be seen as a simple version of the BOXTEP method (refer to Lübbecke and Desrosiers (2005) on the BOXTEP method).

4.5.5 Termination of Column Generation

Consider a case where $Z_\ell^{\mathsf{LP}} > Z_{\mathsf{LB}}$ at Step 5. In this case, a feasible path with negative cost is found for some k. Otherwise, $Z_{\mathsf{LB}} \geq Z_\ell^{\mathsf{LP}} + \sum_{k \in K} r_\ell^k \geq Z_\ell^{\mathsf{LP}}$ when Z_{LB} is updated or not at Step 4. We add Q_ℓ^k to $P_{\ell+1}^k$ for all k, let $\ell := \ell + 1$, and go back to Step 2. In the cases where task sets I_ℓ and $\widehat{I}_\ell$ which are relevant to the constraints (4.9) and (4.11) are updated at Step 4, we also return to Step 2 since the dual solution space is changed.

Since the Lagrangian relaxation problem of (RR^{IP}) at ℓ is also the relaxation problem of (RR_ℓ^{LP}), the other case is that $Z_\ell^{\mathsf{LP}} = Z_{\mathsf{LB}}$. This case indicates that the optimal solution to (RR_ℓ^{LP}) is obtained. We terminate the column generation in this case. If $Z_\ell^{\mathsf{LP}} = M$, then the optimal solution to (RR_ℓ^{LP}), which is a relaxation problem of (RR^{IP}), is $y = 1$ and $x_p^k = 0$ for all $k \in K, p \in P^k$. Hence, (RR^{IP}) is infeasible. We output the message "Rolling stock rescheduling infeasible" and go to the rolling stock uncovered train task detection at Step 7.

4.5.6 Solving Restricted Rolling Stock Rescheduling Problem

At Step 6, we solve the following restricted rolling stock rescheduling problem (RR_ℓ^{IP}) by an optimization solver:

$$
(RR_\ell^{\mathsf{IP}})\quad
\begin{array}{|lrll}
 & \text{minimize} & \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} c_p^k x_p^k + My & \\
 & \text{subject to} & \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} a_{ip}^k x_p^k + y = 1 & \forall i \in I \cup D, \\
 & & \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} a_{\hat{i}p}^k x_p^k + \overline{n_{\hat{i}}}\, y \le \overline{n_{\hat{i}}} & \forall \hat{i} \in \widehat{I}, \\
 & & \displaystyle\sum_{p\in P_\ell^k} x_p^k + y = 1 & \forall k \in K, \\
 & & \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} a_{tp}^{bk} x_p^k + u_t^b y \le u_t^b & \forall b \in B\ \ \forall t \in T, \\
 & & y \ge 0, \quad x_p^k \in \{0,1\} & \forall k \in K\ \ \forall p \in P_\ell^k.
\end{array}
$$

Note that the optimal solution to this problem is not necessarily optimal for (RR^{IP}) since the sets of the feasible paths are different.

If the optimal value of (RR_ℓ^{IP}), denoted by Z_ℓ^{IP}, is less than M, we have the rolling stock rescheduling solution. We output it and stop the algorithm. If $Z_\ell^{\mathsf{IP}} = M$, we output the message "Rolling stock rescheduling solution not found" and go to Step 7.

There may be an optimal or feasible solution to (RR^{IP}) in $P^k \setminus P_\ell^k$. Though it can be found if we apply the branch-and-price method, we believe that it is very cumbersome for us to implement the method and that the implementation is not cost-effective in the rescheduling phase (refer to Barnhart et al. (1998) on branch-and-price).

4.5.7 Solving Rolling Stock Uncovered Train Task Detection Problem

At Step 7, we solve the rolling stock uncovered train task detection (RD^{IP}). We apply column generation to the problem in a similar way to Steps 2–5. Note that we do not

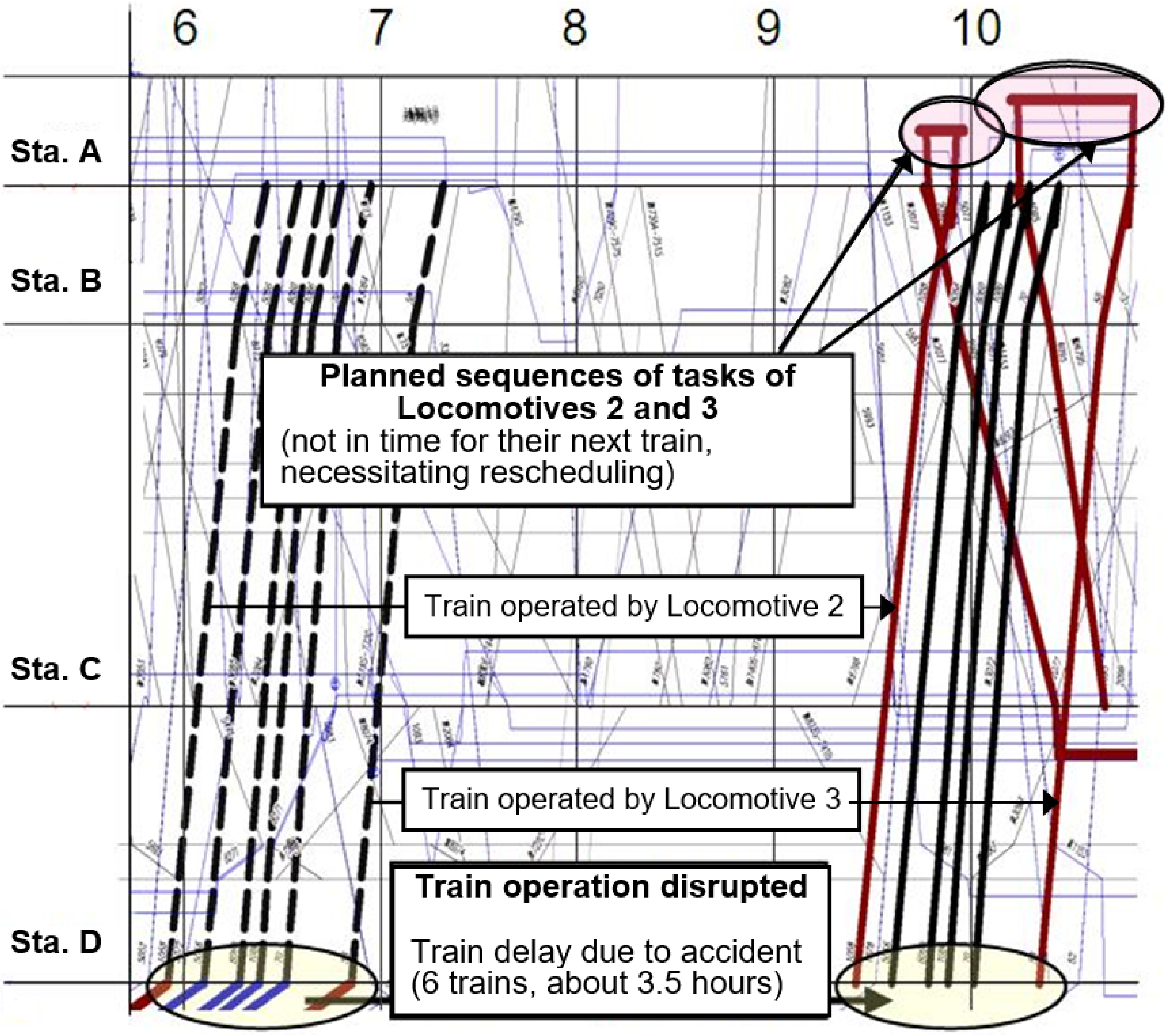

Fig. 4.7 Disruption scenario

relax the constraints on $I \cup D$ and $\widehat{I}$ here.

4.6 Example

We discuss one disruption scenario on freight locomotive on a freight railway network and show the results of freight locomotive rescheduling obtained by the proposed algorithm. Figure 4.7 is a situation in which the departures of six trains at Station D are delayed by about three and a half hours due to an accident. The current sequences of tasks of two Locomotives 2 and 3 are also depicted in the figure, and it indicates that they will miss their next train at Station A because of the delay.

We apply the algorithm to this updated timetable and the current locomotive schedule,

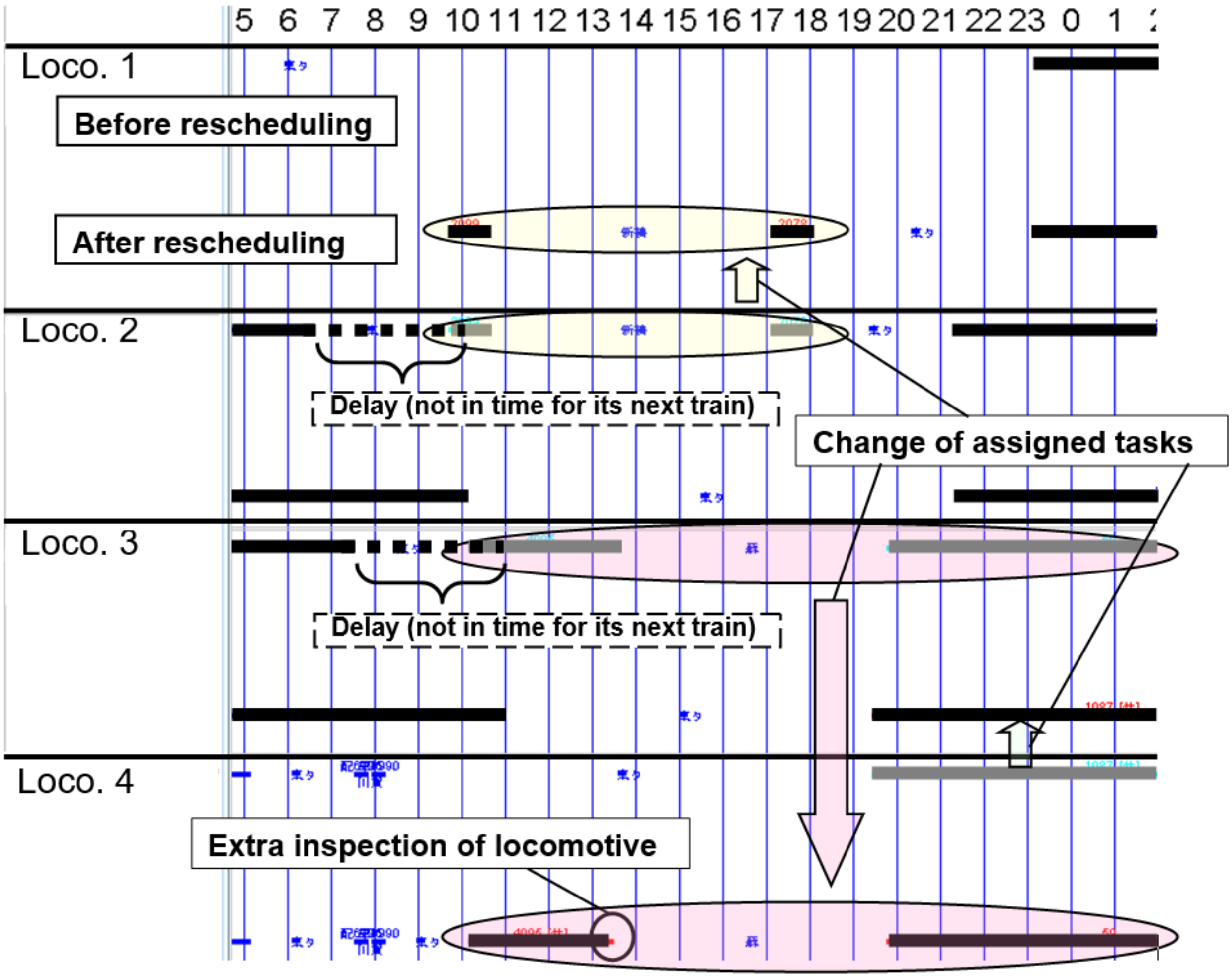

Fig. 4.8 Rolling stock rescheduling solution

and obtain the locomotive rescheduling solution displayed in Figure 4.8. This figure indicates the current sequences of tasks (before rescheduling) and the rescheduled one (after rescheduling) of four locomotives whose sequences are changed by the algorithm. Locomotive 2 is assigned to the train task that Locomotive 1 cannot haul at Station A due to the delay and its next task. The algorithm has exchanged the sequences of tasks of Locomotive 3 for that of Locomotive 4. Extra inspection of Locomotive 4 is carried out since the next inspection time in the current sequence of tasks of Locomotive 3 is later than that of Locomotive 4.

Table 4.4 Disruption cases

No.	# canceled trains	# delayed trains	Average delay (h)	# locos. to miss next trains
1	0	12	3.1	3
2	0	16	2.2	4
3	0	18	3.1	5
4	0	15	4.4	10
5	2	50	3.1	24

4.7 Computational Results

4.7.1 Disruption Cases and Computational Environment

We presents computational results obtained by applying our algorithm to data on the freight locomotives on the freight railway network in Japan shown in Figure 4.9. This network corresponds the region with the heaviest freight train traffic in Japan, where more than 250 freight trains are operated daily. The distance between Kuroiso Station and Shimonoseki Freight Station is about 1,300 km and it takes almost one day for DC electric locomotives to haul the trains between the stations. The area includes the three biggest cities in Japan: Tokyo, Nagoya and Osaka.

We apply five major cases among the real disruption logs from July to November of 2006 which are reported by Japan Freight Railway Company (2006) to the real timetable and locomotive schedule published by Railway Freight Association (2006). They are summarized in Table 4.4. We set $|K| = 144$, which is equal to the number of locomotives planned for the operations on the line, and assume that there is no available reserve locomotive. The locomotives are divided into seven types and their maximum intervals between two consecutive inspections are 72 or 96 hours. The columns of this table show the disruption case number, the number of canceled trains, the number of delayed trains, the average delay time of the delayed trains and the number of locomotives that miss their next trains to haul without rescheduling due to delays or cancellations. Three to 24 (17% of all) locomotives will miss their next trains to haul, and this provides a lower bound of the number of locomotives whose schedules needs to be changed.

We do not know the exact time when the train delay information is reported by Japan Freight Railway Company (2006) and therefore assume that the rescheduling start time of each instance is one hour after the time when the arrival or departure of the trains from the stations is delayed first. We do not pay attention to the rescheduling design deadline for planning a new schedule for any case since we expect that we have rescheduling solution

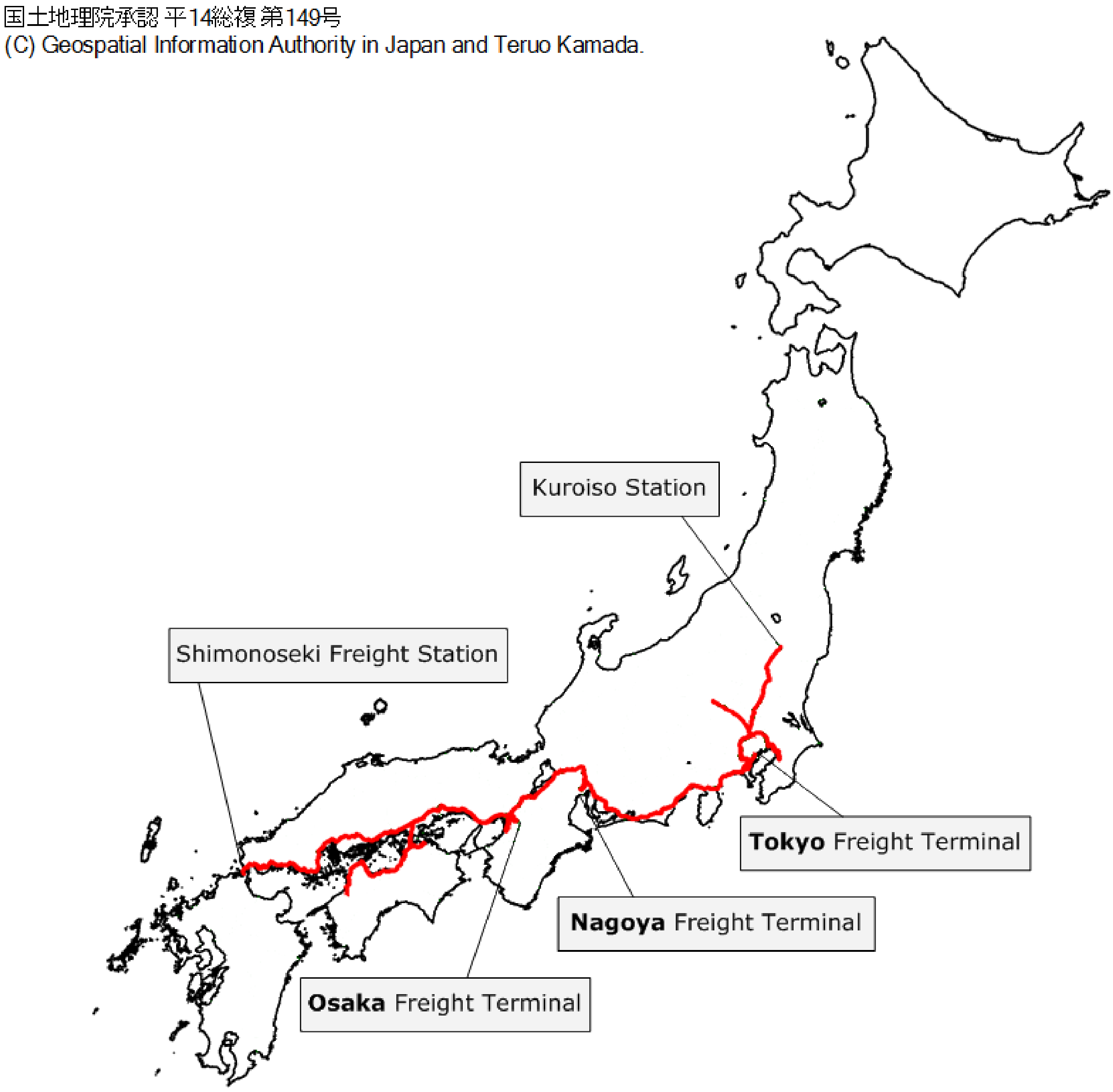

Fig. 4.9 Freight railway network (Geospatial Information Authority in Japan and Kamada (2009))

immediately. We also let the rescheduling period be 36, 48, 60 and 72 hours since there is a task whose running time is over 20 hours. Table 4.5 shows the number of tasks that has to be exactly covered, the number of scheduled and unscheduled inspection vertices, the total number of vertices and edges of the network and the density of the network for each disruption case and each rescheduling period. The density of a network is defined as $|E|/\binom{|V|}{2} \times 100$ where $\binom{|V|}{2} = (|V||V-1|)/2$. The value of $\overline{n_i}$, which is the maximum number of rolling stock units that can be assigned to the deadhead task $\hat{i}$, is two for all $\hat{i} \in \widehat{I}$. We let each time span of the inspections be twelve hours and

Table 4.5 Instance sizes ($|K| = 144$ for every case)

No.	Rescheduling period (h)	$\|I \cup D\|$	$\|S\|$ Scheduled	Unscheduled	$\|V\|$	$\|E\|$	Density (%)
1	36	589	91	210	1,096	17,032	2.84
1	48	763	110	278	1,385	28,518	2.98
1	60	945	140	337	1,673	41,850	2.99
1	72	1,119	159	405	1,962	58,815	3.06
2	36	608	101	209	1,134	18,424	2.87
2	48	784	117	282	1,416	29,665	2.96
2	60	964	150	336	1,711	43,677	2.99
2	72	1,140	166	409	1,993	60,460	3.05
3	36	625	101	220	1,165	19,781	2.92
3	48	796	122	287	1,441	31,031	2.99
3	60	981	150	347	1,742	45,810	3.02
3	72	1,152	171	414	2,018	62,401	3.07
4	36	593	92	210	1,101	17,069	2.82
4	48	767	111	278	1,390	28,561	2.96
4	60	949	141	337	1,678	41,904	2.98
4	72	1,123	160	405	1,967	58,876	3.04
5	36	596	90	211	1,103	17,027	2.80
5	48	770	109	279	1,392	28,523	2.95
5	60	952	139	338	1,680	41,898	2.97
5	72	1,126	158	406	1,969	58,885	3.04

the capacity for each existing rolling stock depot and time span is one plus the number of scheduled inspections in the depot at that time span. The cost parameter values are, $\omega_1 = 100, \omega_2 = 160, \omega_3 = 400, \omega_4 = 180, \omega_5 = 300$ and $M = 30,000$, based on the opinions of experienced workers in the freight locomotive dispatching processes. The importance of each task is given similarly. We set the restoring parameter γ to 0.0, 1.0, 3.0 and 5.0.

The upper limit on the number of feasible paths $|Q^k_\ell|$ added to P^k_ℓ for each k at iteration ℓ is one for $\ell = 1$, 66 for $\ell = 2$ and five otherwise. We impose such limitation since it would take more computation time to solve (RR^{LP}_ℓ) and (RR^{IP}_ℓ) for a larger number of $|P^k_\ell|$. On the other hand, we add relatively many paths when $\ell = 2$. The set P^k_2 consists of only one feasible path selected at $\ell = 1$. After solving (RR^{LP}_2), high dual prices would be set to tasks that were not included in $\bigcup_{k\in K} P^k_2$. We expect that each of such tasks would be included in Q^k_2 for some k by enumerating 66 paths for each locomotive and that all the tasks would be covered for (RR^{LP}_3).

The program is implemented in Java SE 6, calling the Java API of IBM ILOG CPLEX 12.1 (2009) (the current version is offered by IBM (2016)) to solve the linear and integer programming problems. At Step 2 of the algorithm we solve (RR^{LP}_ℓ) by an interior point method, since Vanderbeck (2005) points out that fewer iterations are required for column generation to terminate if an analytic center of the optimal face is adopted as the dual solution (refer to Hillier and Lieberman (2014) on an interior point method). All the experiments are carried out on a 32-bit Windows XP PC having a Core i7-965 CPU (four cores, eight threads, 3.2–3.46 GHz) and 3.5 GB RAM. Four CPU threads are used by the CPLEX and the shortest path problem on eight locomotives is concurrently solved. The algorithm is run ten times for each disruption case and rescheduling period, since feasible paths are enumerated by ($RSP_{\breve{\ell}}$) at some $\breve{\ell}$ in a different order for each trial due to the multi-threading and the dual optimal solution is not unique for ($RR^{\mathsf{LP}}_{\breve{\ell}+1}$), which may cause the linear programming solver to return different values of $\lambda_{i\breve{\ell}+1}, \mu^k_{\breve{\ell}+1}, \nu^b_{t\breve{\ell}+1}$. The number of column generation iterations and therefore the computation time is also expected to differ from trial to trial. We evaluate the maximum running time over the ten trials as well as the average time.

4.7.2 Rescheduling Results

Table 4.6 shows the number of column generation iterations, the total number of enumerated feasible paths and the CPU time required to solve the locomotive rescheduling problem for each restoring parameter value, each disruption case and each rescheduling period. All the results shown in the table are average values over ten trials. In almost all the cases, the numbers of iterations and enumerated paths differ from trial to trial and therefore their averages take fractional values. As we have discussed, it is caused by the multi-threading in solving ($RSP_{\breve{\ell}}$) at some $\breve{\ell}$ and the existence of multiple dual optimal solutions to ($RR^{\mathsf{LP}}_{\breve{\ell}+1}$). Generally, the number of iterations, the number of paths and the CPU time are in proportion to each other. The results are not very different for cases Nos. 1, 2 and 3 or when the rescheduling period is within 48 hours. For cases Nos. 4 and 5 with the rescheduling period being 60 or 72 hours, we obtain a solution in shorter time when we set $\gamma = 3.0$. When $\gamma = 0.0$ or 1.0, we observe that more iterations are required before updating the task sets I_ℓ and $\widehat{I}_\ell$. These sets are frequently updated when $\gamma = 5.0$ while the lower bound is not. In the following experiments, we fix $\gamma = 3.0$.

In Table 4.7, the number of locomotives with a modified sequence of tasks, the objective value of the output integral solution, the MIP gap defined by $(Z^{\mathsf{IP}}_\ell - Z_{\mathsf{LB}})/Z_{\mathsf{LB}} \times 100$, the number of column generation iterations, the total number of enumerated feasible paths and the CPU time (the overall time, the time spent to solve (RR^{LP}_ℓ), (RSP_ℓ) and (RR^{IP}_ℓ) and the maximum of the overall time) are displayed for each disruption case and each rescheduling period. All the results shown in the table are average values over the ten trials except for the maximum CPU time among the trials.

Table 4.6 Restoring parameters and CPU time (average over 10 trials)

γ	No.	Rescheduling period (h)	ℓ	$\sum \lvert P_\ell^k \rvert$	Time (s)	γ	No.	Rescheduling period (h)	ℓ	$\sum \lvert P_\ell^k \rvert$	Time (s)
0.0	1	36	6.0	8,921.0	1.0	1.0	1	36	6.8	8,922.4	1.0
	1	48	7.2	9,530.6	1.4		1	48	8.1	9,564.1	1.6
	1	60	7.2	9,625.2	2.2		1	60	7.8	9,634.4	2.1
	1	72	7.0	9,762.0	3.2		1	72	8.9	9,752.2	4.2
	2	36	19.1	10,640.7	3.4		2	36	21.2	10,480.2	3.7
	2	48	21.8	12,295.1	5.6		2	48	27.5	11,741.9	6.9
	2	60	35.8	14,945.5	16.7		2	60	32.2	14,602.8	15.3
	2	72	25.7	13,505.6	15.8		2	72	27.8	13,675.3	17.2
	3	36	28.5	10,765.2	4.9		3	36	36.3	10,603.0	6.1
	3	48	29.7	13,841.4	9.6		3	48	38.7	13,357.0	11.7
	3	60	26.2	14,574.1	11.8		3	60	26.5	14,803.7	12.3
	3	72	35.6	17,173.7	26.7		3	72	36.4	17,721.1	26.0
	4	36	14.3	9,713.1	2.1		4	36	13.3	9,714.6	2.0
	4	48	19.9	11,125.8	3.8		4	48	23.7	11,266.4	4.6
	4	60	37.1	15,798.3	15.2		4	60	38.5	14,696.4	14.7
	4	72	53.2	19,424.5	36.0		4	72	49.0	19,505.7	32.9
	5	36	3.0	8,728.0	0.5		5	36	3.0	8,728.0	0.5
	5	48	36.2	12,937.7	10.6		5	48	33.8	12,891.2	9.4
	5	60	38.1	15,141.3	17.8		5	60	36.8	14,986.1	24.4
	5	72	45.2	20,650.9	32.4		5	72	42.6	16,733.1	30.4
3.0	1	36	6.0	8,921.3	0.9	5.0	1	36	6.0	8,923.0	0.9
	1	48	7.8	9,546.6	1.6		1	48	8.7	9,549.2	1.8
	1	60	7.1	9,630.1	2.2		1	60	7.0	9,635.4	2.4
	1	72	7.6	9,735.6	3.7		1	72	7.6	9,757.9	3.7
	2	36	22.5	10,340.0	3.8		2	36	26.5	10,186.6	4.5
	2	48	27.6	11,638.6	7.2		2	48	26.5	11,644.3	6.6
	2	60	27.3	12,494.1	12.3		2	60	28.3	11,795.6	11.6
	2	72	23.5	13,597.5	15.3		2	72	25.9	14,448.1	16.6
	3	36	22.3	10,795.2	3.9		3	36	23.2	10,786.5	4.2
	3	48	23.8	12,561.6	6.7		3	48	23.6	12,116.3	6.5
	3	60	26.2	14,850.6	12.1		3	60	31.8	15,364.4	14.7
	3	72	34.6	17,514.2	25.3		3	72	35.4	17,778.5	26.1
	4	36	11.7	9,572.7	1.6		4	36	10.9	9,586.7	1.6
	4	48	18.6	10,898.3	3.7		4	48	22.2	10,959.2	4.8
	4	60	29.0	14,185.5	10.9		4	60	33.1	13,425.1	12.5
	4	72	37.7	17,784.2	23.9		4	72	40.5	18,325.2	27.4
	5	36	3.0	8,728.0	0.5		5	36	3.0	8,728.0	0.5
	5	48	25.2	12,678.5	7.4		5	48	26.1	11,981.9	7.4
	5	60	28.2	16,073.3	14.4		5	60	32.3	18,570.7	18.2
	5	72	37.6	18,215.6	26.7		5	72	47.2	22,716.7	38.3

The instances are feasible except for cases Nos. 3, 4 and 5 with the rescheduling period set to 36 hours. The reason for the rescheduling to become feasible when we set a longer rescheduling period is, that some tasks cannot be covered unless the current sequences for locomotives with different types are exchanged, which has to be done prior to the completion time of the rescheduling. The freight train operator in this area imposes an operational constraint that the type of the locomotive assigned to a convergence task has to be the same as the type assigned to the task in the current locomotive schedule. There

Table 4.7 Rolling stock rescheduling results and CPU time when $\gamma = 3.0$ (average over 10 trials)

No.	Rescheduling period (h)	#locos. with modified tasks	Z_ℓ^{IP}	Gap (%)	ℓ	$\sum \lvert P_\ell^k \rvert$	Time (s)	(RR_ℓ^{LP})	(RSP_ℓ)	(RR_ℓ^{IP})	Max. time (s)
1	36	6.0	1,200.0	0.00	6.0	8,921.3	0.9	0.4	0.5	0.0	1.0
1	48	6.0	1,200.0	0.00	7.8	9,546.6	1.6	0.6	0.9	0.0	1.9
1	60	6.0	1,200.0	0.00	7.1	9,630.1	2.2	0.6	1.5	0.0	2.4
1	72	6.0	1,200.0	0.00	7.6	9,735.6	3.7	0.9	2.9	0.0	4.3
2	36	16.0	5,335.0	0.00	22.5	10,340.0	3.8	2.6	1.2	0.0	4.2
2	48	14.0	5,235.0	0.00	27.6	11,638.6	7.2	4.1	3.1	0.0	8.6
2	60	14.0	5,120.0	1.23	27.3	12,494.1	12.3	5.0	6.2	1.2	17.0
2	72	14.0	4,840.0	0.46	23.5	13,597.5	15.3	4.8	9.2	1.3	18.6
3	36	infeasible	-	-	22.3	10,795.2	3.9	2.6	1.3	0.0	4.9
3	48	18.0	8,430.0	0.00	23.8	12,561.6	6.7	3.7	3.1	0.0	7.7
3	60	16.0	7,850.0	0.00	26.2	14,850.6	12.1	5.4	6.7	0.0	13.4
3	72	17.0	7,710.0	0.00	34.6	17,514.2	25.3	10.7	14.6	0.0	28.0
4	36	infeasible	-	-	11.7	9,572.7	1.6	1.0	0.6	0.0	1.9
4	48	20.4	5,560.0	0.00	18.6	10,898.3	3.7	1.8	1.8	0.0	5.8
4	60	20.8	5,560.0	0.00	29.0	14,185.5	10.9	5.4	5.5	0.0	13.2
4	72	20.4	5,240.0	0.00	37.7	17,784.2	23.9	10.9	13.0	0.0	27.8
5	36	infeasible	-	-	3.0	8,728.0	0.5	0.1	0.4	0.0	0.5
5	48	45.0	16,160.0	0.25	25.2	12,678.5	7.4	3.8	2.4	1.2	9.0
5	60	43.5	15,632.0	1.77	28.2	16,073.3	14.4	6.8	5.2	2.3	16.5
5	72	44.6	14,356.0	0.48	37.6	18,215.6	26.7	11.8	12.4	2.5	29.9

is no time and place for any reexchange in the short rescheduling period while there is a chance for it after 36 hours. Different solutions are obtained from trial to trial for some instances of case No. 4 (when the rescheduling period is 48, 60 or 72 hours) and case No. 5 (60 or 72 hours). The difference of dual solutions returned by the linear programming solver for a particular (RR_ℓ^{LP}) affects the feasible paths enumerated in (RSP_ℓ) and finally the solutions. The number of locomotives with a modified sequence of tasks does not decrease for cases Nos. 3, 4 and 5 though we take a longer rescheduling period. This outcome is possible since we do not directly minimize the number of such locomotives but the sum of rescheduling penalties (divided into the five types). For cases Nos. 1, 4 and 5, there are about twice as many rescheduled locomotives as locomotives that miss their next trains to haul due to the disruption (see Table 4.4), while that does not hold for case No. 2 or 3. Generally, dispatchers in charge of the locomotive rescheduling try to exchange the current sequence of a disrupted locomotive involved in a disruption with an undisrupted one, and our results seem to be comparable to them in this sense. We can examine the relatively small disruption case No. 1, and in fact, our solution is favorably evaluated by experienced workers in the freight locomotive dispatching processes. We observe decrease in of the objective function value when we take a longer rescheduling period for cases Nos. 2, 3, 4 and 5. For case No. 1 with the various rescheduling periods, the same solution

Table 4.8 Rolling stock uncovered train task detection results (average over 10 trials)

No.	Rescheduling period (h)	# uncovered tasks	Gap (%)	ℓ_D	$\sum \lvert P^k_{\ell_D} \rvert$	Time (s)	$(RD^{\mathsf{LP}}_{\ell_D})$	(RLP_{ℓ_D})	$(RD^{\mathsf{IP}}_{\ell_D})$
3	36	3.0	0.01	3.6	411.9	7.0	1.1	0.5	5.4
4	36	1.0	0.00	5.5	705.3	3.0	1.2	0.5	1.3
5	36	12.3	0.01	8.3	2,044.4	4.7	2.8	0.6	1.3

is obtained, in which the locomotive scheduling can be completely recovered in 33 hours.

The MIP gaps shown in Table 4.7 indicate that optimal solutions to the original rescheduling problem (RR^{IP}) are obtained for cases Nos. 1, 3 and 4 regardless of the rescheduling period. We have the optimal solution for case No. 2 with the rescheduling period being 36 and 48 hours. For these cases, the linear programming problem $(RR^{\mathsf{LP}}_{\ell})$ has provided the integral solutions at the last iterations of Steps 2–5. The solutions are very close to optimal for the other cases (the gaps are less than 2%). It should also be noted that the algorithm have stopped at Step 5 for all the cases where the solution is not found. Hence the instances are truly infeasible.

For the cases where the rescheduling is identified as infeasible, the rolling stock uncovered train task detection problem (RD^{IP}) is solved. Table 4.8 shows the number of uncovered tasks, the MIP gap defined in a similar way to the rescheduling problem, the number of column generation iterations (denoted by ℓ_D), the number of enumerated feasible paths and the CPU time. The linear programming problem in the column generation is denoted by $(RD^{\mathsf{LP}}_{\ell_D})$, the longest path problem (since the uncovered train task detection is a maximization problem) by (RLP_{ℓ_D}) and the integer programming problem after the column generation terminates by $(RD^{\mathsf{IP}}_{\ell_D})$. We observe that there exist feasible paths of all the locomotives in every case. Optimal or almost optimal solutions in terms of the objective value of (RD^{IP}) are obtained. The number of uncovered tasks are small for cases Nos. 3 and 4. For case No. 5, there are more uncovered tasks and four different solutions are obtained among the ten trials, which is not large compared to $|I \cup D|$, the number of the tasks which has to be covered.

We present in Table 4.9 the number of locomotives with a modified sequence of tasks, the number of rescheduled locomotives whose sequence of tasks includes a task originally hauled by a different locomotive type (the cost ω_3 of the all cost in the objective value of $(RR^{\mathsf{IP}}_{\ell})$ is relevant to this), the number of unscheduled inspections (the value times cost ω_4 is included in $(RR^{\mathsf{IP}}_{\ell})$) and the number of locomotives assigned to a different convergence task from the current one (the value times ω_5 is also in $(RR^{\mathsf{IP}}_{\ell})$) in the feasible rescheduling solutions to case No. 3. As we take a longer rescheduling period, there are no large changes in the number of rescheduled locomotives assigned to a task originally hauled by a different

Table 4.9 Summary of rescheduling solution to case No. 3

No.	Rescheduling period (h)	# locos. with modified tasks	# locos. assigned to tasks with different type	# unscheduled inspections	# locos. whose current task not assigned at end of rescheduling
3	48	18.0	6	6	10
3	60	16.0	5	7	8
3	72	17.0	7	7	4

type or the number of unscheduled inspections. On the other hand, more locomotives are assigned to their current convergence task at the end of the rescheduling period since there are more opportunities for that and we let ω_5 be relatively large.

4.7.3 Computation Time

From Table 4.7, the number of iterations, the number of paths and the CPU time are generally in proportion to each other. They are also related to the magnitude of disruptions. All the solutions are obtained in ten seconds when we let the rescheduling period 48 hours and within 30 seconds when 72 hours overall, which is acceptable. It takes almost the same time to solve the restricted master problem as the time to solve the column generation subproblem. The integral solutions are found very quickly after the column generation for the cases where (RR_ℓ^{LP}) provides a fractional solution at the last iteration. The computation time to solve (RR_ℓ^{LP}) depends on the size of the network and P_ℓ^k while (RSP_ℓ) on the network size only. The CPU time would not be balanced if our algorithm required more iterations. The difference of the overall computation time among the trials is very small and can be ignored. It should be noted that the short computation time cannot be achieved without our set-covering relaxation; Table 4.10 shows the MIP gap, the number of column generation iterations, the total number of enumerated feasible paths and the overall CPU time for cases Nos. 3, 4 and 5 where $I_0 := I \cup D, \widehat{I}_0 := \widehat{I}$, i.e., the instances are solved without the set-covering relaxation. The MIP gap is scarcely improved for case No. 5. The values of Z_ℓ^{LP} decrease slowly and require more iterations, which cause the enumeration of many feasible paths. Our relaxation approach enhances the computation by a factor of eight at a maximum.

For the cases where the rescheduling is identified as infeasible, Table 4.8 shows that it takes more computation time to solve (RD^{IP}) than to solve (RR^{IP}). Note that the comparison is unfair since there are approximately 10,000 feasible paths enumerated in

Table 4.10 CPU time without set-covering relaxation (average over 10 trials)

No.	Rescheduling period (h)	Gap (%)	ℓ	$\sum \lvert P_\ell^k \rvert$	Time (s)
3	36	-	18.9	11,542.6	4.4
3	48	0.00	30.3	19,295.2	14.6
3	60	0.00	52.1	31,618.7	52.8
3	72	0.00	91.4	52,240.9	213.2
4	36	-	9.7	10,182.5	1.8
4	48	0.00	26.2	16,761.0	9.2
4	60	0.00	49.4	30,262.0	42.3
4	72	0.00	82.0	48,383.0	161.7
5	36	-	3.0	8,728.0	0.5
5	48	0.28	27.5	16,263.9	10.5
5	60	1.76	45.6	25,857.5	33.8
5	72	0.51	68.5	35,357.2	84.2

advance to solve the restricted master problem of (RD^{IP}) and it leads to more computation time. For each of the cases, the restricted master problem provides a fractional solution at the last iteration. The computation time to solve the integer programming problem after the column generation terminates accounts for about 28 to 77 percent of the overall time. This observation indicates the possibility of computational hardness of (RD^{IP}) for large-scale instances.

4.8 Conclusions

We have dealt with the rolling stock rescheduling of passenger or freight trains in this chapter while railway operations on a line or a network are disrupted by disturbance. We have defined the rolling stock rescheduling as to reassign the rolling stock units the train tasks so that the scale of the actions to be rescheduled should be as small as possible and that unscheduled inspections should be as few as possible. Two consecutive inspections of each of the units must not violate the maximum inspection interval permitted. Each train is assumed to consist of one unit. The rolling stock assignment to the tasks involved in the rescheduling period has to be prepared before the rescheduling design deadline comes. We have also defined the detection problem of less important unassigned train tasks called the rolling stock uncovered train task detection if the rolling stock rescheduling has failed.

In the rolling stock rescheduling algorithm, we have constructed the rolling stock rescheduling network from the rescheduling situation. We have then formulated the rolling stock rescheduling as an integer programming problem which can be considered as a variant of the set partitioning problem, and have solved it by column generation. The columns generated in the algorithm have not been enough for us to find an optimal solution to the original integer programming problem. Hence our approach provides optimal or near-optimal solutions depending on problem instances. If no solution has been found, we have formulated the rolling stock uncovered train task detection which can be seen as a variant of the set packing problem. We have introduced set-covering relaxation to enhance the computation. The column generation subproblems with the periodic rolling stock unit inspection constraint have been reduced to the shortest path problems which can be solved in polynomial time. Inspections are carried out once or more in a rescheduled sequence of train tasks for each rolling stock unit if necessary.

We have applied the algorithm to the real disruption cases on the freight railway network with the heaviest freight train traffic in Japan. The optimal or near-optimal solutions have been obtained, depending on the instances. The algorithm has truly decided the infeasible cases when the rescheduling period does not have due length. The number of locomotives whose sequence of tasks modified has been kept small in the solutions. The experienced workers in the freight rolling stock dispatching processes have assessed the solution for the small disruption case and evaluated favorably. The solutions have been obtained in less than 30 seconds for all the cases, which is acceptable in the dispatching processes. Our relaxation approach has enhanced the computation by a factor of eight at a maximum, compared to solving the original problem by column generation without the relaxation.

Chapter 5

Crew Rescheduling

5.1 Introduction

5.1.1 Background

In this chapter, we discuss crew rescheduling of passenger or freight trains while railway operations on a line or a network are disrupted by disturbance, based on (K.) Sato and Fukumura (2011a). Every train, except for automatically operated ones, is operated by a train crew, which is composed of mostly one driver and one or a few conductors. Each crew member has been, on his/her working day, assigned to his/her planned sequence of trains whose operation section and time are specified, which ordinarily begins and ends at the crew base to which he/she belongs. The crew rescheduling is carried out, at a certain point of time, when the railway operations in accordance with the current crew schedule are being, or those to be implemented from the time to the following hours will be, delayed to some extent or unable to be carried out, owing to the disturbance.

Disruption to the crew schedule is caused by disturbance which primarily affect the crew schedule, injury of a crew member on his/her at work for instance, and the rescheduled timetable and/or rolling stock. Examples of the last cases are as follows. If a train is intentionally delayed, to wait for another train on a different line for a connection, or canceled in the timetable rescheduling, the driver of the train may or will be unable to catch his/her next train to operate. If a different type of a rolling stock unit is assigned to a train in the rolling stock rescheduling, there will be a possibility that the driver will have never been trained to operate that type of rolling stock unit and will not be permitted to drive it.

The crew rescheduling plan has to be designed and implemented in real-time, since there is a case where any crew member has not yet been assigned to trains which will depart after the following minutes. Another case is that the disrupted situation might change while we are considering a rescheduling plan and that the plan might not be applicable in the latest situation. The dispatchers in charge of the crew rescheduling have to communicate the modified pieces from the current schedule in the new schedule to the crews via a station or crew base staff, by radio, by phone or by any information technology. This task is cumbersome, hence the scale of the actions to be rescheduled should be as small as possible. Meanwhile, some crew members are obliged to work overtime by their new schedule, which is not desirable. We have to make the rescheduled plan operator-oriented by taking these requests into consideration.

5.1.2 Terminology

On the basis of the disruption management context described above, we define terminology of the crew rescheduling as follows. We deal with the *driver* rescheduling or the *conductor* rescheduling at a time on a railway line or a railway network. We call a certain point of time at which we have noticed necessity for the rescheduling the *current time*. A *disrupted situation* is train, rolling stock and crew operations at the current time which is being or will be disrupted by disturbance. A certain amount of time is required to plan a new schedule and communicate necessary orders to the crews. If part of the crew operations is managed by a certain system (when a driver assistance system has been installed in a train and a server sends the data for instance), the schedule is input to it. We let a *rescheduling design deadline* be a deadline for planning the new schedule. The time when the communication and the input is completed and we can implement the schedule is called *rescheduling start time*. We also set the *rescheduling period*, which is hours from the rescheduling start time.

A timetable and a rolling stock schedule on the line or the network, originally planned in their scheduling phase or rescheduled once or more before the current time in any manner, are given. We call them the *current timetable* and the *current rolling stock schedule*. The *current crew schedule* on the line or the network is also given, which is originally planned or rescheduled once or more before the current time. An *updated timetable* and an *updated rolling stock schedule* are obtained at the current time, which can be equivalent to the current ones, be further disrupted by the disturbance or be rescheduled in timetable and rolling stock rescheduling. We consider a *crew member* to be involved in the rescheduling if he/she is or will be on duty in the rescheduling period in the current crew schedule. A *reserve crew member* is also a crew member if he/she is available. Each of them belongs to a *crew base*.

In any crew schedule, he/she is assigned to his/her sequence of trains whose operation section and time are specified, which ordinarily begins and ends at the crew base to which he/she belongs. The sequence is called a *duty* and a duty in the current crew schedule is called the *current duty*. Each crew member will be ready for the rescheduling at a certain time and place after the rescheduling start time. If he/she is due to go to work during the rescheduling period, he/she will be ready at the time he/she goes to work and at the crew base to which he/she belongs. The trains in each crew member's current duty can be reassigned after the time for him/her be ready for the rescheduling. We call each of them a *train task*. Trains in the rescheduling period to which any crew member has not yet been assigned to are included in the train tasks (they may be divided into several tasks at certain stations) and canceled trains are excluded. If the train of a train task is an out-of-service train, it is called a *deadhead train task*. He/she needs *preparation time* from the end of each train task to the beginning of the next one operated by him/her. He/she has to returns to the crew base at the end of his/her duty. An *importance* value is assigned to each train task, e.g., the importance of deadhead train tasks is zero and that of the rest of tasks depends on the priority of the train.

5.1.3 Approach

We let an updated timetable, an updated rolling stock schedule, the current crew schedule, the disrupted situation, the rescheduling design deadline and the rescheduling start time be given on a railway line or a railway network of passenger or freight trains. The *crew rescheduling* in this chapter is then to reassign the crew members the train tasks so that the scale of the actions to be rescheduled should be as small as possible and that overtime should be as short as possible. In the rescheduling, we have to let each of the members return to the crew base to which he/she belongs to. The number of crew members essential to each train task depends on the characteristics of the train. Ordinarily, the deadhead train tasks require one driver and no conductor, whereas the other tasks require one driver and one or more conductors. A crew member is not permitted to operate a train task if he/she has never been trained to work in the railway section where the train task runs or to operate the rolling stock unit assigned to the train task. The preparation time is necessary between two consecutive train operations by the same crew member. We have to reschedule the crew assignment to the tasks involved in the rescheduling period before the rescheduling design deadline comes.

If the crew rescheduling has failed, i.e., we cannot reassign all the train tasks to the required number of the crew members, timetable and rolling stock rescheduling (which are not restricted to the methods discussed in our timetable rescheduling and rolling stock rescheduling) have to be carried out so that the situation should be resolved. Before that, it is useful to clarify which tasks are unassigned. If we can choose unassigned tasks in the overall train task set, it is desirable that they are of less importance since they will be delayed or canceled in the timetable rescheduling. We call the detection of unassigned train tasks which are less important the *crew uncovered train task detection*.

Firstly we construct the *crew rescheduling network* from the rescheduling situation. We then formulate the crew rescheduling as an integer programming problem which can be considered as a variant of the set covering problem. We solve it by column generation since the all possible duties of the crew members are not known in advance and it will take large amount of time to enumerate them. The columns generated in the algorithm is not enough for us to find an optimal solution to the original integer programming problem. Hence our approach provides near-optimal solutions in general. If no solution is found, we formulate the crew uncovered train task detection as another integer programming problem which can be seen as a variant of the set packing problem. We also solve the problem by column generation. The column generation subproblems are reduced to the shortest path problems. The readers are recommended referring to Hillier and Lieberman (2014) as well as Nemhauser and Wolsey (1999) on a network, Hillier and Lieberman (2014), Nemhauser and Wolsey (1999), Williams (2013) and Wolsey (1998) on integer programming, Nemhauser and Wolsey (1999) on the set covering problem as well as the set packing problem and Desrosiers and Lübbecke (2005) as well as Wolsey (1998) on column generation. Umetani and Yagiura (2007) discusses advanced topics on the set covering problem and Barnhart et al. (1998) on column generation. Other topics on operations research and mathematical optimization are also expounded by Hillier and Lieberman (2014), Nemhauser and Wolsey (1999) and Wolsey (1998).

5.2 Literature Review

There has been a smaller number of crew rescheduling studies than that of timetable rescheduling studies in the rail transport literature. Huisman (2007) deals with crew rescheduling owing to track maintenance in a short-term planning. The problem is formulated as an integer programming formulation of the set covering problem and is solved by column generation. The number of available crews is unfixed and computation time of several hours is acceptable, both of which are not acceptable in our crew rescheduling. Walker et al. (2005) discusses timetable and crew rescheduling simultaneously and minimizes the sum of weighted train idle time and a deviation from target working hours of crew members. The problem is formulated as a mixed integer programming problem, in which crew duties are formulated as the set partitioning problem. A small instance of the problem is solved by row and column generation.

Fujimori et al. (2004) views crew rescheduling as a combinatorial auction problem and solves it by simulated annealing. Kojima et al. (2008) presents a field trial of a crew rescheduling support system, which is based on a tabu search approach by Takahashi et al. (2008). Abbink et al. (2010) presents a method based on multi-agent techniques. In these algorithms, a case where there exists no rescheduling solution is not clearly discussed. An integer multicommodity network flow model which can be applied to rolling stock rescheduling or crew rescheduling is provided by (T.) Sato et al. (2009), and it is solved by partial path exchange heuristics and a local search. Almost the same problem is discussed by (T.) Sato et al. (2012) and it is solved by Lagrangian relaxation. Computational results of the two algorithms are presented on the rolling stock rescheduling only. Flier et al. (2007) discusses a crew duty swap.

Crew rescheduling has been vigorously discussed in the airline industry. Clausen et al. (2010) provides an extensive review of disruption management in the industry. Most of the rescheduling model are based on an integer programming formulation of the set-covering or the set partitioning which is solved by column generation, or an integer multicommodity network flow problem. Clausen (2007) points out the similarity of the disruption management process as well as crew rescheduling models and algorithms between the railway and airline industry. On the other hand, it also indicates the difference in the problem scale and the computation time limit. In the maritime industry, crew rescheduling is not important since crew members always follow a vessel and do not have work rules, according to Brouer et al. (2013). Huisman and Wagelmans (2006) presents dynamic and simultaneous scheduling of vehicles and crew on a road network. Bus crew rescheduling is reviewed and studied by Shibghatullah (2008). It should be noted that, in general, crew members can move between any two points on a road network, which does not hold true on a railway networks.

In our crew rescheduling, the fixed current timetable is given, since the simultaneous optimization is intractable in practical cases in Japan or the timetable rescheduling may be done a different entity. In the railway industry, our crew rescheduling is one of the first approaches that formulate the rescheduling as an integer programming problem which can be considered as a variant of the set covering or the set partitioning problem. The other, independent approaches are by Potthoff et al. (2010) and Rezanova and Ryan (2010). The former is based on the set covering problem and the latter the set partitioning problem. All of the three models are solved by column generation. Recall that our algorithm does not necessarily provide an optimal solution to every instance due to a limited number of columns to be generated. It provides an optimal solution or decides the infeasibility of the instance correctly, however, when the column generation terminates with an integral solution.

5.3 Problem Description

5.3.1 Rescheduling Network

We construct the crew rescheduling network $\mathcal{N} := (V, E, c, f)$ from the rescheduling situation on a railway line or a railway network on which we focus on. The notation is summarized in Table 5.1. Then rescheduling start time is h_0 and the rescheduling period is h. The set of vertices V is defined as $V := K \cup I \cup \widehat{I} \cup B$. A set of the crew members is denoted by K, and crew member k is in K if

$$\texttt{Start\_time}(k) \leq h_0 \text{ and } \texttt{End\_time}(k) \geq h_0, \text{ or } h_0 \leq \texttt{Start\_time}(k) \leq h_0 + h$$

where $\texttt{Start\_time}(k)$ and $\texttt{End\_time}(k)$ are the current start and the end time of k's duty, respectively. For each $k \in K$, $\texttt{Base}(k)$ is the crew base to which k belongs to, the time ready for the rescheduling is $\texttt{Ready\_time}(k)$, and the station where k is at $\texttt{Ready\_time}(k)$ is $\texttt{Ready\_sta}(k)$. We let I be the set of trains tasks and $\widehat{I}$ be deadhead train tasks. It consists of the train tasks in the current duty of each $k \in K$ which have not yet been operated at $\texttt{Ready\_time}(k)$ or canceled. Train tasks in the rescheduling period to which any crew member has not yet been assigned are also included in I (they may be divided into several tasks at certain stations). An element i of I or $\widehat{I}$ has its departure time $\texttt{Dep\_time}(i)$, its departure station $\texttt{Dep\_sta}(i)$, its arrival time $\texttt{Arr\_time}(i)$ and its arrival station $\texttt{Arr\_sta}(i)$. The number of required crew members of $i \in I \cup \widehat{I}$ is denoted by $\underline{n_i}$. Ordinarily, $\underline{n_i} = 1$ for all $i \in I \cup \widehat{I}$ in the driver rescheduling, whereas $\underline{n_i} \geq 1$ for $i \in I$ and $\underline{n_{\hat{i}}} = 0$ for $\hat{i} \in \widehat{I}$ in the conductor rescheduling. Preparation time $\texttt{Prep\_time}(i_1, i_2)$ are required between pairs of train tasks (which include deadhead train tasks) $(i_1, i_2) \in (I \cup \widehat{I}) \times (I \cup \widehat{I})$. We call B the set of crew bases. Let S_b consists of the set of adjacent stations to $b \in B$ and the crew base b itself.

We next define the directed edge set E. The first type of the edges is drawn from a crew member to a train task. For $(k, i) \in K \times (I \cup \widehat{I})$, the pair is an element of E if the following condition is satisfied:

$$\texttt{Ready\_sta}(k) = \texttt{Dep\_sta}(i) \text{ and } \texttt{Ready\_time}(k) \leq \texttt{Dep\_time}(i).$$

For pair $(k, b) \in K \times B$, if

$$\texttt{Ready\_sta}(k) \in S_b$$

Table 5.1 Notation for crew rescheduling network

$\mathbb{R}_+$	set of nonnegative real numbers
h_0	rescheduling start time
h	rescheduling period
$\mathcal{N} := (V, E, c, f)$	crew rescheduling network
$V := K \cup I \cup \widehat{I} \cup B$	set of vertices
K	set of crew members
`Start_time`(k)	current start time of duty of $k \in K$
`End_time`(k)	current end time of duty of $k \in K$
`Base`(k)	crew base to which $k \in K$ belongs
`Ready_time`(k)	time ready for rescheduling for $k \in K$
`Ready_sta`(k)	station or base where $k \in K$ is at `Ready_time`(k)
I	set of train tasks
$\widehat{I}$	set of deadhead train tasks
`Dep_time`(i)	departure time of $i \in I \cup \widehat{I}$
`Dep_sta`(i)	departure station of $i \in I \cup \widehat{I}$
`Arr_time`(i)	arrival time of $i \in I \cup \widehat{I}$
`Arr_sta`(i)	arrival station of $i \in I \cup \widehat{I}$
$\underline{n_i}$	min. number of crew members necessary for $i \in I \cup \widehat{I}$
`Prep_time`(i_1, i_2)	preparation time required between $(i_1, i_2) \in (I \cup \widehat{I}) \times (I \cup \widehat{I})$
B	set of crew bases
S_b	set of adjacent stations to $b \in B$ and b
E	set of directed edges
$c : K \times E \to \mathbb{R}_+$	rescheduling cost of $e \in E$ for $k \in K$
ω_1	positive cost value of assignment of different crew member's duty who belongs to same crew base
ω_2	positive cost value of assignment of different crew member's duty who belongs to different crew base
`Overtime`$(*)$	overtime cost function on time
ω_3	positive coefficient of overtime cost
$f : I \cup \widehat{I} \to \mathbb{R}_+$	importance value of $i \in I \cup \widehat{I}$
P^k	set of feasible paths for $k \in K$ on $\mathcal{N}$
c_p^k	rescheduling cost of $p \in P^k$ for $k \in K$, defined as $c_p^k := \cup_{e \in p \cap E} c_e^k$
f_p^k	importance value of $p \in P^k$ for $k \in K$, defined as $f_p^k := \cup_{i \in p \cap (I \cup \widehat{I})} f_i$

is satisfied, (k, b) is included in E. Let us consider pair of train task vertex pair (including deadhead train tasks) $(i_1, i_2) \in (I \cup \widehat{I}) \times (I \cup \widehat{I})$. We let $(i_1, i_2) \in E$ if

$$\texttt{Arr\_sta}(i_1) = \texttt{Dep\_sta}(i_2)$$
$$\text{and } \texttt{Arr\_time}(i_1) + \texttt{Prep\_time}(i_1, i_2) \leq \texttt{Dep\_time}(i_2)$$

holds. When

$$\texttt{Arr\_sta}(i) \in S_b$$

is satisfied for $(i, b) \in (I \cup \widehat{I}) \times B$, we let $(i, b) \in E$.

Figure 5.1 is an example of a time-space diagram of the current timetable of Trains 1M–6M and 8M and duties of Drivers 1 and 2 on a railway line. The circles shown in the figure indicate the beginning of the drivers' duties, and the triangles show the end of the duties. Driver 1 is at Station B and Driver 2 is at Station C at the current time. Suppose here that the departure of Train 2M at Station C is updated and delayed for minutes, to wait for passengers from another railway line for instance. Figure 5.2 shows this situation, which will cause Driver 2 to miss Train 4M unless the driver duty is changed. In this case, no extra train delay will occur if Driver 1 is instructed to operate 4M and the following train tasks and Driver 2 is instructed to operate 1M and the following. Note that their duties must be changed again after the arrival of Train 3M at Station C or Train 8M at Station B, since they belong to different crew bases. Figure 5.3 is the crew rescheduling network constructed from Figure 5.2. The vertices of this network consist of the crew members (k_1 and k_2), the tasks to be covered (from i_1 to i_6 as well as $i_8^{\texttt{A}}, i_8^{\texttt{B}}$ where the superscripts A and B mean the arrival stations), and the crew bases ($b_{\texttt{A}}$ and $b_{\texttt{C}}$). The directed edges are drawn between the vertices if the constraint on the time and the station is satisfied.

Let $\mathbb{R}_+$ be a set of nonnegative real numbers, and the cost function $c : K \times E \to \mathbb{R}_+$ returns a nonnegative value when a crew members and an edge are input. For simplicity, we substitute c_e^k for $c(k, e)$, and it means the cost for crew member k to traverse edge e. We set $c_e^k := 0$ for edge $e \in E \cap ((K \cup I \cup \widehat{I}) \times (I \cup \widehat{I}))$ when its endpoint tasks coincide with a crew members and his/her next task or two consecutive train tasks in the current duty of any crew member. Some positive cost is set otherwise, since it indicates a change of crew member assignment. We prepare two cost values ω_1 and ω_2 with $\omega_1 < \omega_2$. When the edge connects the train tasks of two crew members who belong to the same crew base, we select ω_1. Otherwise, we do ω_2. A different positive value is added to edge $(v_1, v_2) \in E \cap ((K \cup I \cup \widehat{I}) \times (I \cup \widehat{I}))$ for crew member k when the connection causes

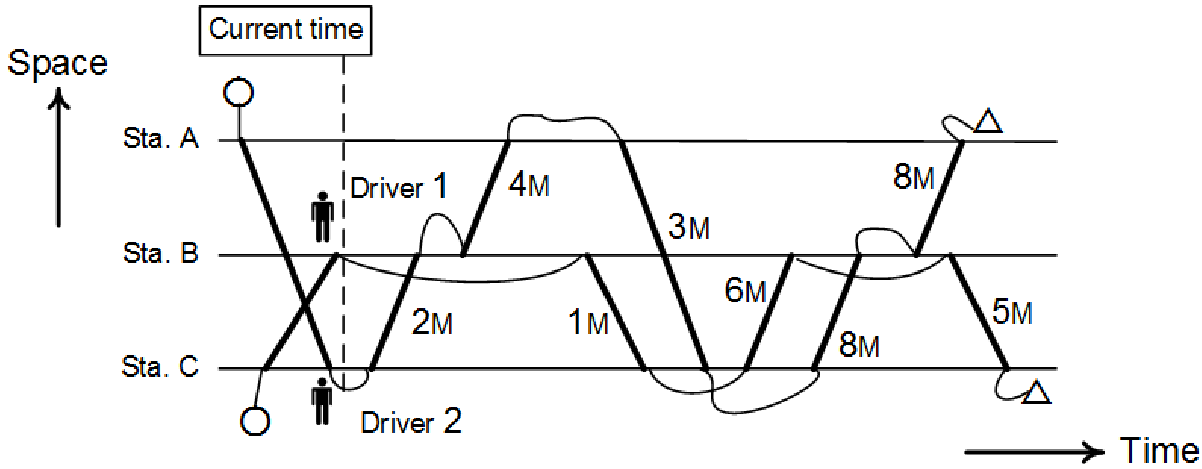

Fig. 5.1 Time-space diagram of current timetable and crew duties

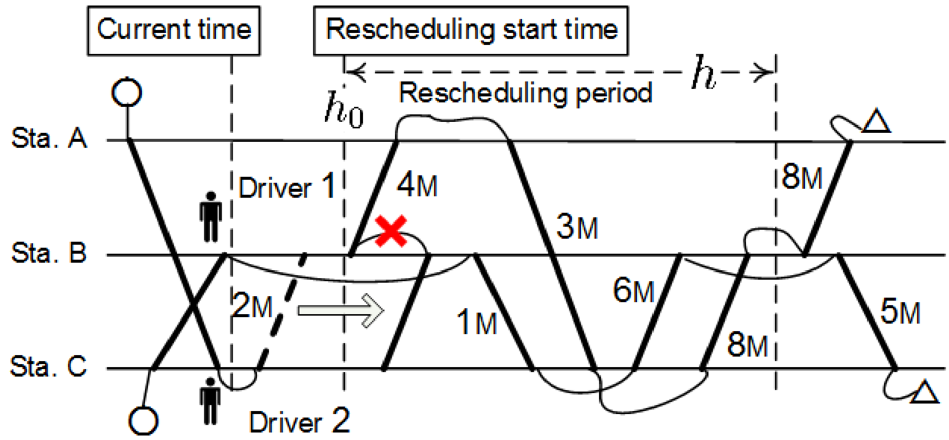

Fig. 5.2 Updated timetable

overtime. Specifically, we set

$$c^k_{(v_1,v_2)} := c^k_{(v_1,v_2)} + \begin{cases} \texttt{Overtime}(\texttt{Arr\_time}(v_2)) & \text{if } v_1 \in K, \\ \texttt{Overtime}(\texttt{Arr\_time}(v_2)) - \texttt{Overtime}(\texttt{Arr\_time}(v_1)) & \text{otherwise} \end{cases}$$

according to the following overtime cost function:

$$\texttt{Overtime}(\alpha) := \begin{cases} 0 & \text{if } \alpha \leq \texttt{End\_time}(k), \\ \omega_3 \times \left(\dfrac{(\alpha - \texttt{End\_time}(k))\ (\mathrm{m})}{60} \right)^2 & \text{otherwise} \end{cases}$$

where ω_3 is a positive parameter (refer also to Figure 5.4). For $(v, i) \in E \cap ((K \cup I \cup \widehat{I}) \times$

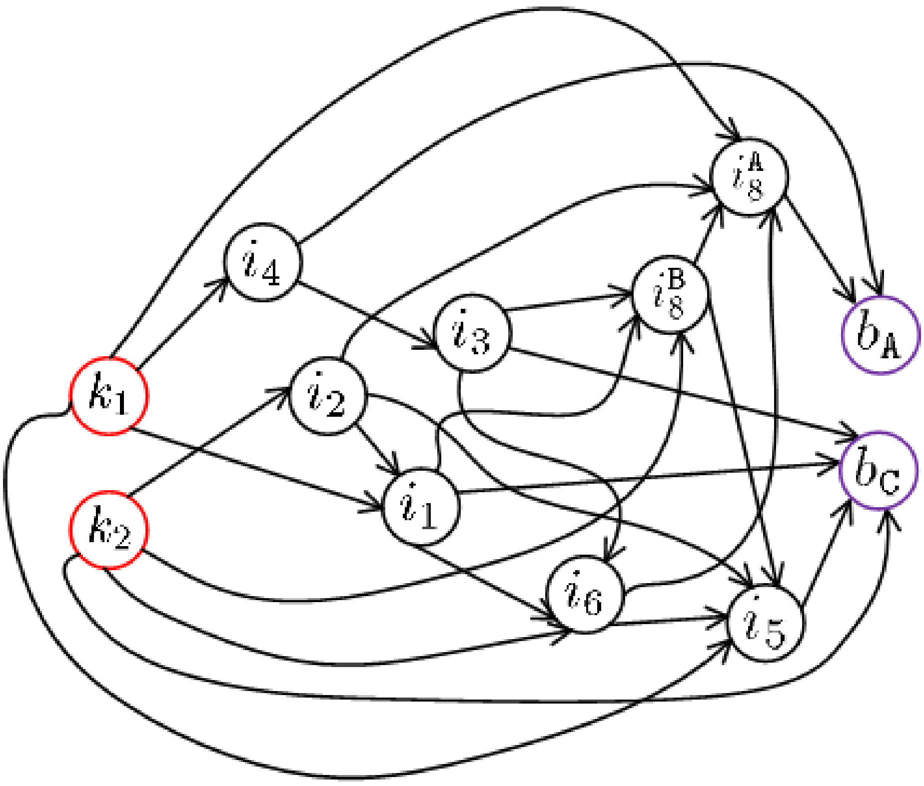

Fig. 5.3 Crew rescheduling network

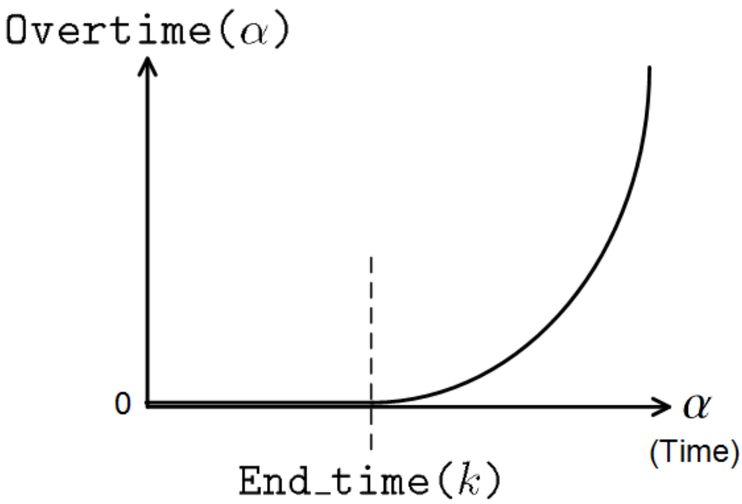

Fig. 5.4 Overtime cost function

$(I \cup \widehat{I})$), if driver k is not permitted to operate i, we let $c_e^k := \infty$. Let $f : I \cup \widehat{I} \to \mathbb{R}_+$ be importance value of $i \in I \cup \widehat{I}$.

5.3.2 Feasible Paths

The cost of path p of driver $k \in K$ is denoted by c_p^k and it is defined as $c_p^k := \cup_{e \in p \cap E} c_e^k$. Then p with $c_p^k = 0$ is equivalent to the current duty of k without overtime. A bigger value of c_p^k means that degree of the modification and/or overtime is large. Similarly, the importance of path p of driver $k \in K$ is denoted by f_p^k and it is defined as $f_p^k := \cup_{i \in p \cap (I \cup \widehat{I})} f_i$.

Path p of driver $k \in K$ is called a *feasible path* if p starts at k, $c_p^k < \infty$ and ends at the crew base to which k belongs to. We let P^k as the set of feasible paths for $k \in K$.

5.3.3 Problem Definitions

The rescheduling network $\mathcal{N}$ is given, and we define the crew rescheduling as to assign each crew member $k \in K$ one feasible path $p \in P^k$ and to cover each of the all train tasks $i \in I \cup \widehat{I}$ by the required number $\underline{n_i}$ of feasible paths so that the sum of the cost of the assigned path c_p^k for all k is minimized. If it is not possible, we solve the crew uncovered train tasks detection. It is defined as to maximize the sum of the importance of tasks f_p^k covered by p of all k, whose necessary conditions are that each k is assigned to one feasible path or no path and that each i is not covered by more than $\underline{n_i}$ of feasible paths. The former constraint is to make the problem feasible in a case where the set of feasible paths is empty for a certain crew member. The latter prevents an important task to be covered by many paths.

5.4 Integer Programming Formulations

5.4.1 Crew Rescheduling

We formulate the crew rescheduling as an integer programming problem. The notation is summarized in Table 5.2. We let constant a_{ip}^k be

$$a_{ip}^k := \begin{cases} 1 & \text{if train task } i \in I \cup \widehat{I} \text{ is included in feasible path } p \in P^k \text{ of crew} \\ & \text{member } k \in K, \\ 0 & \text{otherwise.} \end{cases}$$

Table 5.2 Notation for crew rescheduling formulations and algorithm

Constants	
c_p^k	refer to Table 5.1
f_p^k	refer to Table 5.1
a_{ip}^k	1 if $i \in I \cup \widehat{I}$ is included in $p \in P^k$ for $k \in K$, 0 otherwise
n_i	refer to Table 5.1
$\overline{M}$	arbitrary large number
Decision Variables	
x_p^k	1 if $k \in K$ is assigned to $p \in P^k$, 0 otherwise
y	dummy variable
Temporary Sets and Values	
ℓ	iteration counter of column generation
P_ℓ^k	set of feasible paths for $k \in K$ on $\mathcal{N}$ at ℓ
$\lambda_{i\ell}$	dual price of $i \in I \cup \widehat{I}$ at ℓ
μ_ℓ^k	dual price of $k \in K$ at ℓ
Q_ℓ^k	set of feasible paths to be added for $k \in K$ on $\mathcal{N}$ at ℓ
r_ℓ^k	feasible shortest path length for $k \in K$ at ℓ
Formulations and Objective Values	
(CR^{IP})	crew rescheduling problem
Z_{LB}	lower bound value of (CR^{IP})
(CD^{IP})	crew uncovered train task detection problem
(CR_ℓ^{LP})	restricted master problem of (CR^{IP}) at ℓ
Z_ℓ^{LP}	objective value of (CR_ℓ^{LP}) at ℓ
(CSP_ℓ)	column generation subproblem of (CR^{IP}) at ℓ
(CR_ℓ^{IP})	restricted crew rescheduling problem at ℓ
Z_ℓ^{IP}	objective value of (CR_ℓ^{IP})

We then let x_p^k a decision variable defined as

$$x_p^k := \begin{cases} 1 & \text{if crew member } k \in K \text{ is assigned to feasible path } p \in P^k, \\ 0 & \text{otherwise.} \end{cases}$$

The notation is given, and we formulate the crew rescheduling as an integer programming problem which can be considered as a variant of the set covering problem with a set-partitioning side constraint. The formulation is shown below and we call it (CR^{IP}):

$$(CR^{\mathsf{IP}}) \quad \begin{array}{|lllr} & \text{minimize} & \sum_{k\in K}\sum_{p\in P^k} c_p^k x_p^k & & (5.1)\\ & \text{subject to} & \sum_{k\in K}\sum_{p\in P^k} a_{ip}^k x_p^k \geq \underline{n_i} & \forall i \in I \cup \widehat{I}, & (5.2)\\ & & \sum_{p\in P^k} x_p^k = 1 & \forall k \in K, & (5.3)\\ & & x_p^k \in \{0,1\} & \forall k \in K \;\; \forall p \in P^k. & (5.4) \end{array}$$

The objective function (5.1) is the total cost of feasible paths to which the crew members are assigned, and we try to minimize it. The constraint (5.2) means that all the train tasks, including the deadhead train tasks, have to be covered by $\underline{n_i}$ or more feasible paths of the crew members. When $\underline{n_i} = 1$, it is the set covering constraint. Each crew member is assigned to one feasible path in his/her feasible path set, which is stated in the constraint (5.3). The constraint can be seen as set-partitioning of K. The constraint (5.4) restrict the value of the decision variables to either zero or one. We let Z^{IP} be the objective value of (CR^{IP}).

5.4.2 Crew Uncovered Train Task Detection

We call the integer programming formulation of the crew uncovered train task detection problem (CD^{IP}). It is shown below:

$$(CD^{\mathsf{IP}}) \quad \begin{array}{|lll} & \text{maximize} & \sum_{k\in K}\sum_{p\in P^k} f_p^k x_p^k & \\ & \text{subject to} & \sum_{k\in K}\sum_{p\in P^k} a_{ip}^k x_p^k \leq \underline{n_i} & \forall i \in I \cup \widehat{I},\\ & & \sum_{p\in P^k} x_p^k \leq 1 & \forall k \in K,\\ & & x_p^k \in \{0,1\} & \forall k \in K \;\; \forall p \in P^k. \end{array}$$

The constraint on $I \cup \widehat{I}$ is called the set-packing constraint. Note that this problem is always feasible since $x_p^k = 0$ for all $k \in K$ and $p \in P^k$ satisfies the constraints, provided that we define the sum of x_p^k is zero when $P^k := \varnothing$. Any train task $i \in I \cup \widehat{I}$ that is covered by less than $\underline{n_i}$ paths in a solution means that the task cannot be operated due to the lack of crew members. The solution to (CD^{IP}) thus helps us decide which train should be delayed or canceled in timetable rescheduling.

Step 1: *Initialization.*
Set $\ell := 1, Z_{\mathsf{LB}} := 0, P^k_\ell := \varnothing \ \forall k \in K$.
Step 2: *Solving Restricted Master Problem.*
Solve (CR^{LP}_ℓ).
Let Z^{LP}_ℓ be objective value, $\lambda_{i\ell}, \mu^k_\ell$ be dual prices.
Step 3: *Solving Column Generation Subproblem.*
Let $-\lambda_{i\ell}, -\mu^k_\ell$ be cost on corresponding vertices of $\mathcal{N}$.
Parallel for $k \in K$,
  Solve shortest path problem on $\mathcal{N}$.
  Let r^k_ℓ be feasible shortest path length.
  Let Q^k_ℓ be set of feasible paths with negative cost.
Step 4: *Lower Bound Update.*
Set $Z_{\mathsf{LB}} := \max\{Z_{\mathsf{LB}},\ Z^{\mathsf{LP}}_\ell + \sum_{k\in K} r^k_\ell\}$.
Step 5: *Termination of Column Generation.*
If $Z^{\mathsf{LP}}_\ell > Z_{\mathsf{LB}}$, then
  Set $P^k_{\ell+1} := P^k_\ell \cup Q^k_\ell \ \forall k \in K, \ell := \ell + 1$ and Go to Step 2.
Else if $Z^{\mathsf{LP}}_\ell = M$, then
  Output “Crew rescheduling infeasible.” and Go to Step 7.
Step 6: *Solving Restricted Crew Rescheduling Problem.*
Solve (CR^{IP}_ℓ).
Let Z^{IP}_ℓ be objective value.
If $Z^{\mathsf{IP}}_\ell < M$, then
  Apply deadheading selection.
  Output crew rescheduling solution and Stop.
Else,
  Output “Crew rescheduling solution not found.”
Step 7: *Solving Crew Uncovered Train Task Detection Problem.*
Solve (CD^{IP}) by column generation.
Output crew uncovered train task detection solution and Stop.

Fig. 5.5 Crew rescheduling algorithm

5.5 Algorithm

5.5.1 Overall Algorithm and Initialization

We apply column generation to (CR^{IP}) since all the elements of the feasible path set P^k for each $k \in K$ are not known in advance, and it will take large amount of time to enumerate them. The overall algorithm is displayed in Figure 5.5.

At Step 1 of the algorithm, we let ℓ be an iteration counter of the column generation and set $\ell := 1$. We also let Z_{LB} be a lower bound value of (CR^{IP}) and set $Z_{\mathsf{LB}} := 0$. A subset of P^k for each $k \in K$ at ℓ is denoted by P^k_ℓ and we set $P^k_1 := \varnothing$.

5.5.2 Solving Restricted Master Problem

Let M be arbitrary large number and y be a nonnegative dummy variable. We also let x^k_p be a nonnegative continuous variable. We define the restricted master problem (CR^{LP}_ℓ) of (CR^{IP}) as

$$
(CR^{\mathsf{LP}}_\ell) \quad
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{k\in K}\sum_{p\in P^k_\ell} c^k_p x^k_p + My & \quad (5.5)\\
\text{subject to} & \displaystyle\sum_{k\in K}\sum_{p\in P^k_\ell} a^k_{ip} x^k_p + \underline{n_i} y \geq \underline{n_i} \quad \forall i \in I \cup \widehat{I}, & \quad (5.6)\\
& \displaystyle\sum_{p\in P^k_\ell} x^k_p + y = 1 \quad \forall k \in K, & \quad (5.7)\\
& y \geq 0, \quad x^k_p \geq 0 \quad \forall k \in K \ \ \forall p \in P^k_\ell. & \quad (5.8)
\end{array}
$$

This is a linear programming problem (refer to Hillier and Lieberman (2014), Nemhauser and Wolsey (1999) and Williams (2013) on linear programming), and is always feasible if we let $y = 1$ and $x^k_p = 0$ for all $k \in K, p \in P^k$.

At Step 2, we solve (CR^{LP}_ℓ) by using an optimization solver. We let Z^{LP}_ℓ be the objective value of (CR^{LP}_ℓ), $\lambda_{i\ell}$ be the dual price corresponding to train task i of the constraint set (5.6) and μ^k_ℓ be that corresponding to the crew member k of the constraint set (5.7).

5.5.3 Solving Column Generation Subproblem

After (CR^{LP}_ℓ) is solved to optimality, we decide whether the objective value Z^{LP}_ℓ can further be improved by adding new feasible paths to the set P^k_ℓ for any crew member at Step 3. We search for such a feasible path by modeling the column generation subproblem (CSP_ℓ) shown below, which is derived from the dual of (CR^{LP}_ℓ) and the linear relaxation of (CR^{IP}):

$$
(CSP_\ell) \quad
\begin{array}{ll}
\text{find} & k \in K \ \ p \in P^k \\
\text{such that} & c^k_p - \displaystyle\sum_{i\in I\cup\widehat{I}} a^k_{ip}\lambda_{i\ell} - \mu^k_\ell < 0.
\end{array}
$$

We can reduce the search problem to the shortest path problem from $k \in K$ to `Base(k)` since c^k_p is defined as the sum of the cost of the edges included in p and the dual prices $\lambda_{i\ell}, \mu^k_\ell$ relate to the vertices of the network $\mathcal{N}$. Moreover, the shortest path of one crew member is independent of that of another. We therefore let $-\lambda_{i\ell}, -\mu_\ell^k$ the cost of the corresponding vertices of $\mathcal{N}$ and solve the shortest path problems concurrently for several of K on the network.

The shortest path problem for any crew member k is solved in $O(|I| + |\widehat{I}| + |E|)$ by Dijkstra's algorithm on an acyclic graph. The order of vertices to be visited is, the crew member vertex k, the train tasks in ascending order of `Arr_time(`i`)`, and the base to which k belongs. Refer to Hillier and Lieberman (2014) as well as Nemhauser and Wolsey (1999) on the shortest path problem as well as Dijkstra's algorithm and Wolsey (1998) on those on a directed acyclic graph.

Let r_ℓ^k be feasible shortest path length for k (we set $r_\ell^k := 0$ when $P^k = \varnothing$). We add feasible paths with negative cost to set Q_ℓ^k if they exist for some k. The feasible paths other than the shortest one which are obtained by Dijkstra's algorithm can also be added.

5.5.4 Lower Bound Update

The shortest feasible paths found at ℓ is equal to the optimal solution to the Lagrangian relaxation problem of (CR^{IP}) with the constraint (5.2) being relaxed, if we regard the dual price $\lambda_{i\ell}$ as a Lagrangian multiplier (refer to Desrosiers and Lübbecke (2005), Nemhauser and Wolsey (1999) and Wolsey (1998) on Lagrangian relaxation and duality). The formulation is as follows:

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{k\in K}\sum_{p\in P^k} c_p^k x_p^k + \sum_{i\in I\cup\widehat{I}} \lambda_{i\ell}\Big(\underline{n_i} - \sum_{k\in K}\sum_{p\in P^k} a_{ip}^k x_p^k\Big) & \\
\text{subject to} & \displaystyle\sum_{p\in P^k} x_p^k = 1 & \forall k \in K, \\
& x_p^k \in \{0,1\} & \forall k \in K \ \ \forall p \in P^k.
\end{array}
$$

Let $\widetilde{x}_p^k$ be one if it is the shortest feasible path p of the crew k, and be zero otherwise, which is equivalent to the optimal solution to the Lagrangian problem. By the duality theorem, $r_\ell^k \leq 0$ for every k and the following holds:

$$
\begin{aligned}
&\sum_{k\in K}\sum_{p\in P^k} c_p^k \tilde{x}_p^k + \sum_{i\in I\cup\hat{I}} \lambda_{i\ell}(\underline{n_i} - \sum_{k\in K}\sum_{p\in P^k} a_{ip}^k \tilde{x}_p^k) \\
&= \sum_{k\in K}\sum_{p\in P^k} c_p^k \tilde{x}_p^k + \sum_{i\in I\cup\hat{I}} \underline{n_i}\lambda_{i\ell} - \sum_{k\in K}\sum_{p\in P^k}\sum_{i\in I\cup\hat{I}} a_{ip}^k \lambda_{i\ell}\tilde{x}_p^k + \sum_{k\in K}\mu_\ell^k - \sum_{k\in K}\mu_\ell^k \sum_{p\in P^k}\tilde{x}_p^k \\
&= \sum_{i\in I\cup\hat{I}} \underline{n_i}\lambda_{i\ell} + \sum_{k\in K}\mu_\ell^k + \sum_{k\in K}\sum_{p\in P^k}(c_p^k - \sum_{i\in I\cup\hat{I}} a_{ip}^k\lambda_{i\ell} - \mu_\ell^k)\tilde{x}_p^k \\
&= Z_\ell^{\mathsf{LP}} + \sum_{k\in K} r_\ell^k.
\end{aligned}
$$

That is, the optimal value of the Lagrangian relaxation problem is equal to the objective value of (CR_ℓ^{LP}) plus the sum of the shortest path lengths. Since this value is a lower bound of (CR^{IP}), we can update Z_{LB} if $Z_{\mathsf{LB}} < Z_\ell^{\mathsf{LP}} + \sum_{k\in K} r_\ell^k$ at Step 4.

5.5.5 Termination of Column Generation

Consider a case where $Z_\ell^{\mathsf{LP}} > Z_{\mathsf{LB}}$ at Step 5. In this case, a feasible path with negative cost is found for some k. Otherwise, $Z_{\mathsf{LB}} \geq Z_\ell^{\mathsf{LP}} + \sum_{k\in K} r_\ell^k \geq Z_\ell^{\mathsf{LP}}$ when Z_{LB} is updated or not at Step 4. We add Q_ℓ^k to $P_{\ell+1}^k$ for all k, let $\ell := \ell + 1$, and go back to Step 2.

Since the Lagrangian relaxation problem of (CR^{IP}) at ℓ is also the relaxation problem of (CR_ℓ^{LP}), the other case is that $Z_\ell^{\mathsf{LP}} = Z_{\mathsf{LB}}$. This case indicates that the optimal solution to (CR_ℓ^{LP}) is obtained. We terminate the column generation in this case. If $Z_\ell^{\mathsf{LP}} = M$, then the optimal solution to (CR_ℓ^{LP}), which is a relaxation problem of (CR^{IP}), is $y = 1$ and $x_p^k = 0$ for all $k \in K, p \in P^k$. Hence, (CR^{IP}) is infeasible. We output the message "Crew rescheduling infeasible" and go to the crew uncovered train task detection at Step 7.

5.5.6 Solving Restricted Crew Rescheduling Problem

At Step 6, we solve the following restricted crew rescheduling problem (CR_ℓ^{IP}) by an optimization solver:

$$
(CR_\ell^{\mathsf{IP}}) \quad \left| \quad
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} c_p^k x_p^k + My & \\
\text{subject to} & \displaystyle\sum_{k\in K}\sum_{p\in P_\ell^k} a_{ip}^k x_p^k + \underline{n}_i y \geq \underline{n}_i & \forall i \in I \cup \widehat{I}, \\
& \displaystyle\sum_{p\in P_\ell^k} x_p^k + y = 1 & \forall k \in K, \\
& y \geq 0, \quad x_p^k \in \{0,1\} & \forall k \in K \ \ \forall p \in P_\ell^k.
\end{array}
\right.
$$

Note that the optimal solution to this problem is not necessarily optimal for (CR^{IP}) since the sets of the feasible paths are different.

If the optimal value of (CR_ℓ^{IP}), denoted by Z_ℓ^{IP}, is less than M, we have the crew rescheduling solution. When more than $\underline{n}_i$ crew members are assigned to $i \in I \cup \widehat{I}$, we impose a rule that members which are not assigned to i in their current duty and have early `End_time(`k`)` will deadhead. After this deadhead selection is done, if necessary, we output it and stop the algorithm. If $Z_\ell^{\mathsf{IP}} = M$, we output the message "Crew rescheduling solution not found" and go to Step 7.

There may be an optimal or feasible solution to (CR^{IP}) in $P^k \setminus P_\ell^k$. Though it can be found if we apply the branch-and-price method, we believe that it is very cumbersome for us to implement the method and that the implementation is not cost-effective in the rescheduling phase (refer to Barnhart et al. (1998) on branch-and-price).

5.5.7 Solving Crew Uncovered Train Task Detection Problem

At Step 7, we solve the crew uncovered train task detection (CD^{IP}). We apply column generation to the problem in a similar way to Steps 2–5.

5.6 Example

We discuss one disruption scenario on drivers on a freight railway network and show the results of driver rescheduling obtained by the proposed algorithm. Figure 5.6 is a situation in which one train is canceled and the departures of three trains at Station U are delayed by about three and a half hours due to an accident. The current duty of Driver 3 is also depicted in the figure, and it indicates that he/she will miss his/her next train at Station X because of the delay.

We apply the algorithm to this updated timetable and the current driver duty, and obtain the driver rescheduling solution displayed in Figure 5.7. This figure indicates the current duties (before rescheduling) and the rescheduled duties (after rescheduling) of four

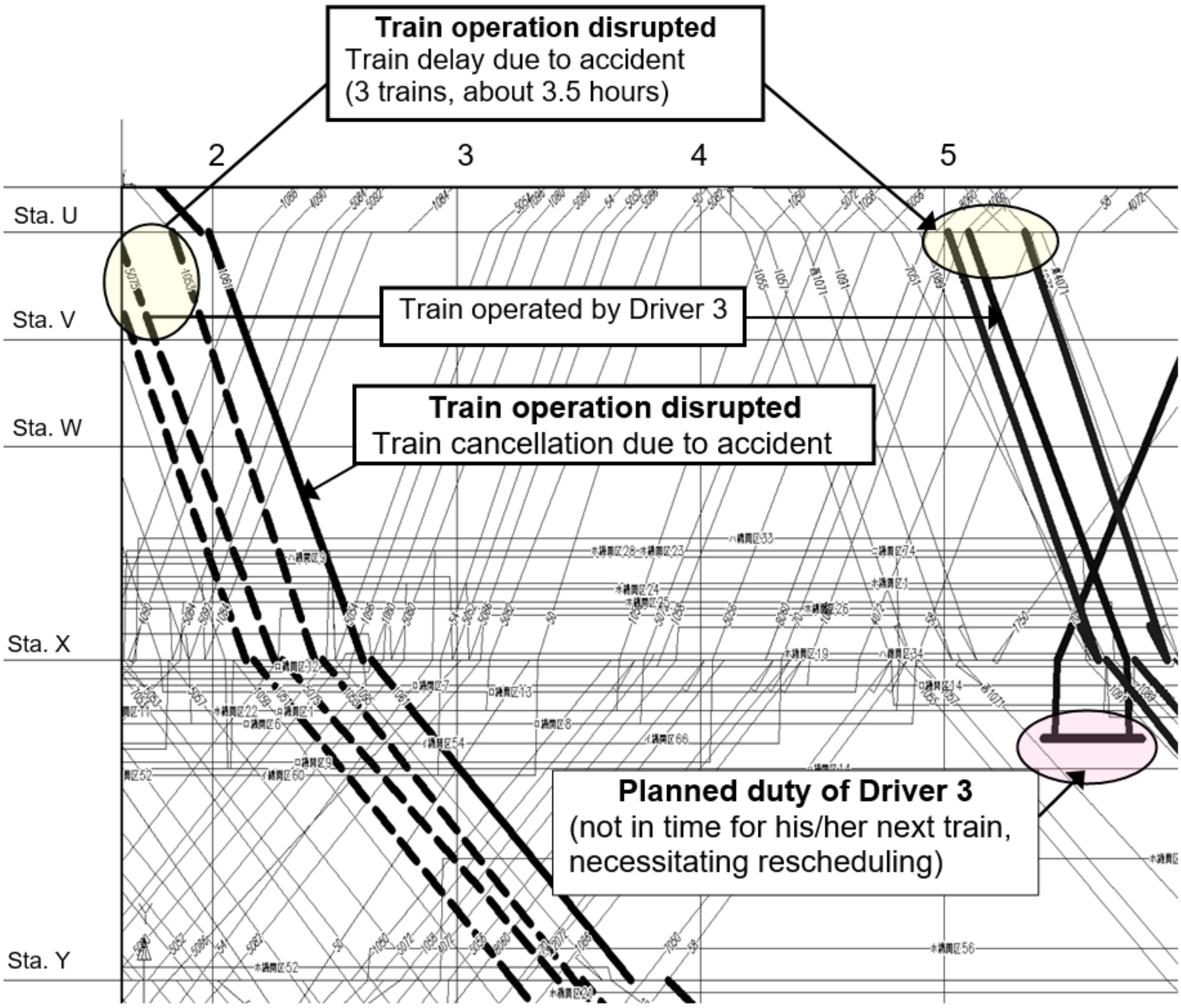

Fig. 5.6 Disruption scenario

drivers whose duties are changed by the algorithm. Driver 1 is assigned to the train tasks that Driver 3 is due to drive at Station X, and Driver 2 is assigned to the two tasks due to be operated by Driver 1 . Driver 4 moves to Station X by deadheading because of the cancellation of the train that he/she was due to drive, while he/she operates the rest of the trains to which he/she is assigned in his/her current duty.

5.7 Computational Results

5.7.1 Disruption Cases and Computational Environment

We present computational results obtained by applying our algorithm to data on the drivers on the freight railway network in Japan shown in Figure 5.8. This network corre-

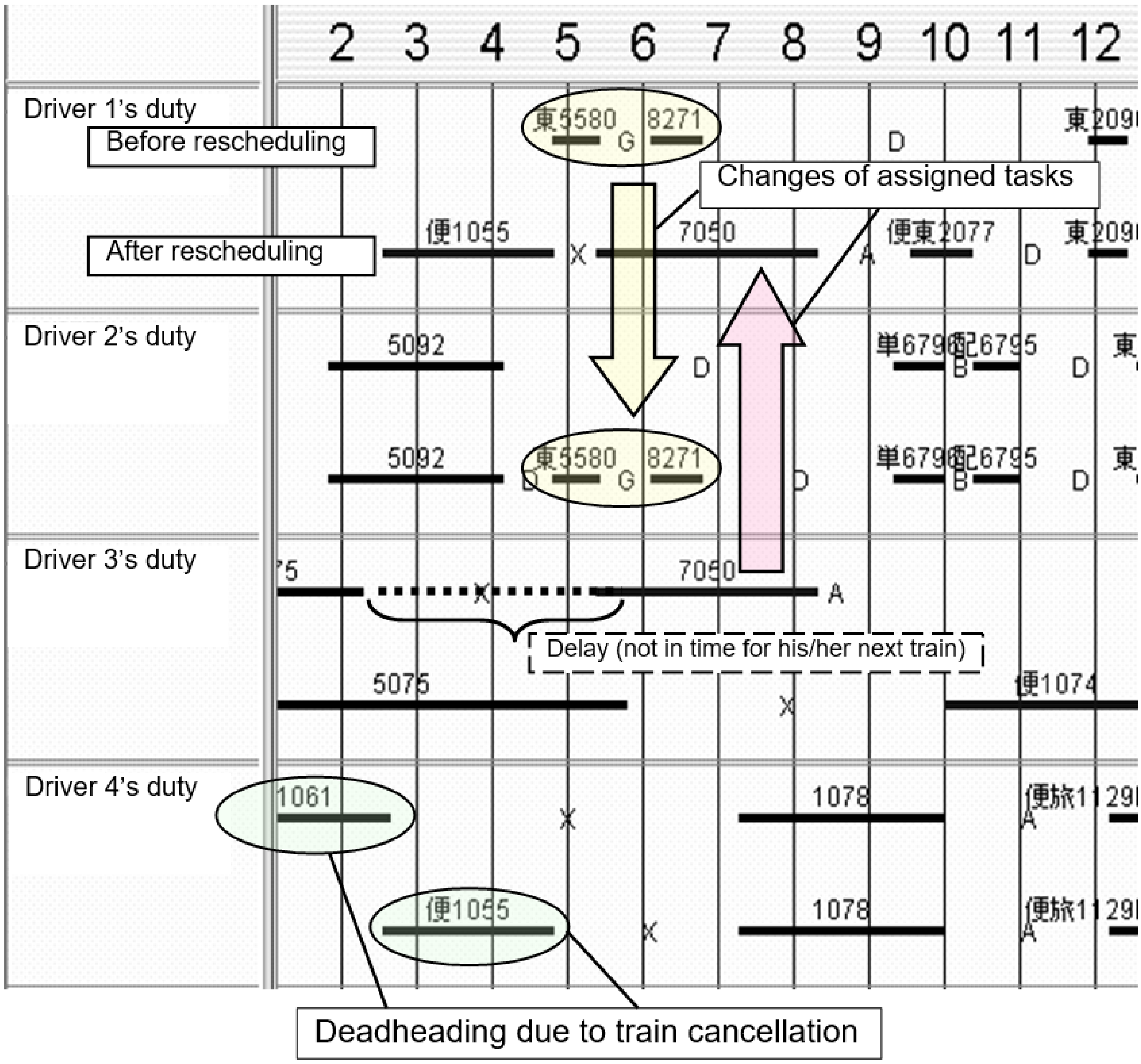

Fig. 5.7 Driver rescheduling solution

sponds to the main area of the region with the heaviest freight train traffic in Japan and more than 200 freight trains are operated daily.

We apply three major cases among the real disruption logs from July to November of 2006 which are reported by Japan Freight Railway Company (2006) as well as two arbitrary disruptions, to the real timetable published by Railway Freight Association (2006) and the driver duties. They are summarized in Table 5.3. The arbitrary disruption cases are A4 and A5 in the table. We do not know the exact time when the train delay information is reported by Japan Freight Railway Company (2006) and therefore assume that the rescheduling start time of each instance is one hour after the time when the arrival or departure of the trains from the stations is delayed first. We do not pay attention to the

Fig. 5.8 Freight railway network (Geospatial Information Authority in Japan and Kamada (2009))

Table 5.3 Disruption cases

No.	# canceled trains	# delayed trains	Average delay (h)	# duties to miss next trains
1	0	11	2.0	4
2	7	25	4.0	5
3	1	29	4.5	7
4A	6	13	4.0	15
5A	11	35	6.0	34

rescheduling design deadline for planning a new schedule for any case since we expect that we have rescheduling solution immediately. We let the rescheduling period be 24 hours. We assume here that no reserve driver is available. Then the number of drivers involved in the rescheduling is 165 ($= |K|$) to which belong to four different crew bases in the largest case. The columns of Table 5.3 show the disruption case number, the number of canceled trains, the number of delayed trains, the average delay time of the delayed trains and the number of driver duties that miss their next trains to operate without rescheduling due to delays or cancellations. Four to 34 drivers will miss their next trains, and this provides a lower bound of the number of drivers whose schedule needs to be changed. This provides a lower bound of the number of drivers whose schedule need to be changed.

Table 5.4 Crew rescheduling results and CPU time

No.	# changed duties	# duties with overwork	Total overtime (h)	Time (s)
1	8	13	12.1	13.0
2	12	10	13.8	15.7
3	12	7	14.2	12.3
4A	23	7	19.0	23.0
5A	54	24	14.2	22.0

The number of train tasks involved in the rescheduling is 727 ($= |I| + |\widehat{I}|$) in the largest case. The trains are operated by one driver, hence $\underline{n_i} = 1$ for all $i \in I \cup \widehat{I}$ and the constraint (5.2) is the set-covering constraint. There are 23 `Dep_sta(*)` and `Arr_sta(*)` stations of the train tasks. The cost parameter values are, $\omega_1 = 100, \omega_2 = 140, \omega_3 = 40$ and $M = 100,000$, based on the opinions of experienced workers in the freight driver dispatching processes.

The upper limit on the number of feasible paths $|Q^k_\ell|$ added to P^k_ℓ for each k at iteration ℓ is one for $\ell = 1$, 66 for $\ell = 2$ and five otherwise. We impose such limitation since it would take more computation time to solve (CR^{LP}_ℓ) and (CR^{IP}_ℓ) for a larger number of $|P^k_\ell|$. On the other hand, we add relatively many paths when $\ell = 2$. The set P^k_2 consists of only one feasible path selected at $\ell = 1$. After solving (CR^{LP}_2), high dual prices would be set to tasks that were not included in $\bigcup_{k \in K} P^k_2$. We expect that each of such tasks would be included in Q^k_2 for some k by enumerating 66 paths for each driver and that all the tasks would be covered for (CR^{LP}_3).

The program is implemented in Java SE 5, calling the Java API of ILOG CPLEX 10.1 (2006) (the current version is offered by IBM (2016)) to solve the linear and integer programming problems. The linear programming problems are solved by the primal simplex method (refer to Hillier and Lieberman (2014) as well as Nemhauser and Wolsey (1999) on the primal simplex method). All the experiments are carried out on a 32-bit Windows XP PC having a Core 2 Duo E7300 CPU (two cores, two threads, 2.66 GHz) and 3.5 GB RAM. One CPU thread is used by the CPLEX and the shortest path problem on two drivers is concurrently solved.

5.7.2 Rescheduling Results

Table 5.4 the summary of the rescheduling solutions. The driver rescheduling solutions are found for all the cases even though we assume that no reserve driver is available. For each disruption case, the number of changed duties, the number of duties with overtime and the sum of overtime are displayed.

The number of duties to miss next trains shown in Table 5.3 is the minimum number of drivers whose duty has to be changed. The number of changed duties in the solutions is close to twice as many as the minimum value for all the cases. Generally, dispatchers in charge of the driver rescheduling try to exchange a duty involved in a disruption with an undisrupted one, and our results seem to be comparable or even preferable to them in this sense. In fact, our solutions for cases Nos. 1, 2 and 3 are favorably evaluated by experienced workers in the freight driver dispatching processes. The number of duties with overtime is small in contrast to the number of delayed trains. Long overtime to one particular driver is avoided by the overtime cost function.

For cases Nos. 1, 2 and 3, the solutions to (CR_ℓ^{LP}) when the algorithm terminates are integral solutions. Although the objective value Z_ℓ^{LP} when the algorithm terminates is fractional for case No. 4A, the objective value of (CR_ℓ^{IP}) is the ceiling of Z_ℓ^{LP}. Since we let the cost of all the edges of the network be an integer (the overtime cost function value is rounded), the truly optimal solution is obtained also for this case. For case No. 5A, a fractional optimal solution is obtained when the algorithm terminates, and the integral solution with the same objective value is obtained.

5.7.3 Computation Time

The solutions are obtained in less than 25 seconds for all the cases. The computation time is acceptable in the dispatching processes. We analyze why the solutions are obtained in such short time. According to Umetani and Yagiura (2007), the set covering problem can be solved in shorter time when the range of the coefficients of the variables in the objective function is large. This holds true for our crew rescheduling problem. The primal simplex method to solve the restricted master problem works very well; the optimal solution to $(CR_{\ell+1}^{\mathsf{LP}})$ is obtained after a small number of iterations from the optimal solution to (CR_ℓ^{LP}). Furthermore, as we have described above, the solutions to (CR_ℓ^{LP}) when the algorithm terminates are integral solutions for cases Nos. 1, 2 and 3. For the other cases, the gap between the optimal values of (CR_ℓ^{IP}) and (CR_ℓ^{LP}) is small or even zero. It should be noted, however, that there is no theoretical property on the integral optimality of the linear relaxation problems of the crew rescheduling problem.

5.8 Conclusions

This chapter has provided the crew rescheduling of passenger or freight trains while railway operations on a line or a network are disrupted by disturbance. We have dealt with the driver or conductor rescheduling at a time. The crew rescheduling in this chapter has been defined as to reassign the crew members the train tasks so that the scale of the actions to be rescheduled should be as small as possible and that overtime should be as short as possible. In the rescheduling, we have to let each of the members return to the crew base to which he/she belongs to. Each train task requires a minimum number of crew members. The crew assignment to the tasks involved in the rescheduling period has to be prepared before the rescheduling design deadline comes. We have also defined the detection problem of less important unassigned train tasks called the crew uncovered train task detection if the rolling stock rescheduling has failed.

In the crew rescheduling algorithm, we have constructed the crew rescheduling network from the rescheduling situation. We have then formulated the crew rescheduling as an integer programming problem which can be considered as a variant of the set covering problem, and have solved it by column generation. The columns generated in the algorithm have not been enough for us to find an optimal solution to the original integer programming problem. Hence our approach provides optimal or near-optimal solutions depending on problem instances. If no solution has been found, we have formulated the crew uncovered train task detection which can be seen as a variant of the set packing problem. The column generation subproblems have been reduced to the shortest path problems.

We have applied the algorithm to the real disruption cases as well as the arbitrary ones on the freight railway network with heavy freight train traffic in Japan. The optimal solutions have been obtained for all the cases, though it does not necessarily holds due to the limited number of columns to be generated. The number of changed duties and the overtime have been small in the solutions. For the real cases, they have been favorably evaluated by the experienced workers in the freight driver dispatching processes. The solutions have been obtained in less than 25 seconds for all the cases, which is acceptable in the dispatching processes.

Chapter 6

Concluding Remarks

6.1 Conclusions

Rail transport systems in the greater part of Japan and the world at the present day, which are considerably large and highly complex, are operated in accordance with a set of predetermined schedules. The rail transport scheduling consists mainly of the timetable scheduling, the rolling stock scheduling, the crew scheduling and the shunting scheduling. When railway operations planned in any manner are disrupted to a certain extent by disturbance, the dispatchers inevitably reschedule the plans to manage the situation. The rail transport rescheduling is a real-time process, and that makes the task more strenuous. In this thesis, we have discussed optimization approaches to the timetable rescheduling, the rolling stock rescheduling and the crew rescheduling in the rail transport rescheduling, to assist the dispatchers in their disruption management. We have providing integer programming formulations and algorithms for them.

In the timetable rescheduling, the train operations and the passengers' behavior are simultaneously modeled. We have first modeled and solved the arrival delay minimization problem. We have next introduced some flexibility in the delay-minimized timetable and solved the passenger inconvenience minimization problem. Numerical results based on the medium-sized railway line and delay contracted from the actual line with heavy train traffic have indicated the trade-off between the total amount of delay and the total amount of increased inconvenience. We have obtained a better solution in terms of the inconvenience at the expense of a minimum delay timetable by introducing flexibility in the delay-minimized timetable. The results have also indicated the other trade-off between obtaining a better solution by enlarging the solution space and the computation time. The computation could last until the rescheduling design deadline comes.

In the rolling stock rescheduling algorithm, we have introduced set-covering relaxation to enhance the computation. The column generation subproblems with the periodic rolling stock unit inspection constraint have been reduced to the shortest path problems which can be solved in polynomial time. Inspections are carried out once or more in a rescheduled sequence of train tasks for each rolling stock unit if necessary. We have applied the algorithm to the real disruption cases on the freight railway network with the heaviest freight train traffic in Japan. The optimal or near-optimal solutions have been obtained, depending on the instances. The algorithm has truly decided the infeasible cases when the rescheduling period does not have due length. The number of locomotives whose sequence of tasks modified has been kept small in the solutions. The experienced workers in the freight rolling stock dispatching processes have assessed the solution for the small disruption case and evaluated favorably. The solutions have been obtained in less than 30 seconds for all the cases, which is acceptable in the dispatching processes. Our relaxation approach has enhanced the computation by a factor of eight at a maximum, compared to solving the original problem by column generation without the relaxation.

In the crew rescheduling algorithm, we have formulated the crew rescheduling as an integer programming problem which can be considered as a variant of the set covering problem, and have solved it by column generation. We have applied the algorithm to the real disruption cases as well as the arbitrary ones on the freight railway network with heavy freight train traffic in Japan. The optimal solutions have been obtained for all the cases, though it does not necessarily holds due to the limited number of columns to be generated. The number of changed duties and the overtime have been small in the solutions. For the real cases, they have been favorably evaluated by the experienced workers in the freight driver dispatching processes. The solutions have been obtained in less than 25 seconds for all the cases, which is acceptable in the dispatching processes.

These results indicate applicability of mathematical optimization to the practical rescheduling situations. By the passenger-oriented approach, compared to the train-punctuality-oriented approach, the deterioration of the service level to the passengers will be smaller. The heavy burden of the dispatchers will is relieved by the operator-oriented rolling stock and crew rescheduling approaches.

6.2 Future Work

6.2.1 Timetable Rescheduling

The applicability of our model to railway lines with higher frequency of train operations has to be discussed. The lower bound of the linear programming relaxation of the formulation is not sufficiently strong, and this makes the optimization difficult and time consuming. An iterative approach is promising which simulates the passengers' behavior in a tentative rescheduled timetable and changes the timetable based on the simulated passengers' behavior, though it might not provide an optimal or even near-optimal solution. Another idea would be to apply row generation, since only a small part of the constraints is ordinarily active at an optimal solution.

In our rescheduling model, we have omitted the capacity issue of a train. At stations in and around Tokyo, the passengers, particularly in the morning on weekdays and during large disruption, cannot board a train and are obliged to wait for the next train due to the congestion. We have to take the situation into account if we try to solve such instances.

We have optimized the sum of further inconvenience to the passengers. This is equivalent to minimize the mean of the increased inconvenience distribution (or the histogram) of the passengers. Train operations as public services have to reduce railway users who suffer much inconvenience, even though it increases the inconvenience to the other users. Optimization of the inconvenience distribution is what is to be discussed.

We also have to deal with the situations where various trains are operated. There are trains which are irrelevant to the passengers and have a certain priority, such as freight trains and out-of-service trains. This indicates that different optimization criteria have to be coordinated.

6.2.2 Rolling Stock Rescheduling

We have assumed in the rolling stock rescheduling that only one rolling stock unit is assigned to a train and distinguished the coupling and uncoupling of units as well as the routing inside the depots and the stations from the rescheduling. A flow model of rolling stock units will be easier to be handled, as it is done in several studies that we have reviewed. On the other hand, the current path model connects each unit with its sequence of tasks and is hence suitable for modeling the periodic inspection. The combination of the two models will be one of future approaches.

6.2.3 Crew Rescheduling

Although we have included overtime of a train crew member in the objective function, his/her workload in his/her working hours is not discussed. Recall that his/her duty is generated in the column generation subproblem and that the subproblem is reduced to the simple shortest path problem. We would have to add constraints for a rest, a meal, etc., to the shortest path problem and solve the constrained path problem. The inspection in the rolling stock scheduling can be viewed as the rest or meal break in the crew rescheduling, then the network extension approach performed in the rolling stock rescheduling will be effective.

6.2.4 Shunting Rescheduling

We have not yet proposed any integer programming or other model to deal with the shunting rescheduling. In the timetable rescheduling, we have modeled arrival and departure orders of the trains. In other words, we have formulated the ordering problem between the trains that run on or stop at the same track segment. We have also modeled the local rerouting, which can be seen as the track assignment to the trains. These decisions corresponds to occupation of a track by a rolling stock unit and its shunting movement. As we have just stated, an integer multicommodity flow model is promising for us to handle coupling and uncoupling. The combination of these formulations will be the whole shunting model.

Some of the existing studies in the literature that we have reviewed will also be applicable. If the shunting is not feasible in accordance with the rescheduled timetable, rolling stock circulation and crew duties, the preceding rescheduling has to be designed again. An algorithm to give a hint for the whole rescheduling to be made feasible will be required which is similar to the rolling stock and the crew uncovered train task detection,

6.3 Prospects for Practical Real-time Rescheduling

Since the instances that the rescheduling algorithms presented in this thesis have been applied to are limited, thorough field trials will have to be performed if a rescheduling support system in which the algorithms are installed is to be introduced to an actual railway line or an actual railway network. At its first step, for safety reasons, it should not be connected to the existing systems, the train traffic control system for instance, in a train traffic control center. A standalone computer will be installed instead and the dispatchers will see a proposed schedule as reference.

The dispatchers will first have the information of an initial delay and they will also play the sole role to input the delay. The other input data, mainly the originally planned schedules, have to be in the rescheduling system. Even through the originally planned schedules are saved as electric data nowadays, it should be noted that some data have not yet been available in an electrical format. There is a possibility that the last inspection date and time of a rolling stock unit is recorded on a paper list. Another case is that the sections of a railway network in which a driver can operate trains are not recorded at all. We must identify these data and save them. In the timetable rescheduling, the predicted amount of the initial delay and passenger OD data are necessary. As we have reviewed, the studies on these topics exist. We have to continue to pay attention to the latest results. Although all this data collection is independent of the cores of the rescheduling algorithms, we have to overcome the difficulty to perform comprehensive assessment of the algorithms and to pave the way for their practical use.

Suppose that the field trials successfully ends and that the rescheduling support system is put into practical use. The system would provide the rescheduling which is passenger-oriented with light dispatchers' burdens if operated in conjunction with other systems such as the train traffic control system and a state-of-the-art communication system to tell the relevant staff the changes in the schedules. On the other hand, the introduction of such new systems would require more or less change in the current workflow of the rescheduling process, and the train operator might not accept this. Therefore, the rescheduling system has also to be ready to be installed in the current context of rescheduling process. Whichever the case may be, we have to put as much effort as we have done to study the rescheduling algorithms into the realization of their practical use.

Bibliography

Abbink, E. J. W., Mobach, D. G. A., Fioole, P. J., Kroon, L. G., van der Heijden, E. H. T. and Wijngaards, N. J. E. (2010). Real-time train driver rescheduling by actor-agent techniques. *Public Transport*, vol. 2, no. 3, pp. 249–268. doi:10.1007/s12469-010-0033-6.

Abe, K. and Araya, S. (1985). Optimization of train order by the branch and bound method. *Transactions of the Society of Instrument and Control Engineers*, vol. 21, no. 5, pp. 514–521. doi:10.9746/sicetr1965.21.514 (in Japanese).
安部恵介, 荒屋真二 (1985). 分枝限定法を用いた列車順序最適化手法. 計測自動制御学会論文集, vol. 21, no. 5, pp. 514–521. doi:10.9746/sicetr1965.21.514.

Abe, K. and Araya, S. (1986). Train traffic simulation using the longest path method. *IPSJ Journal*, vol. 27, no. 1, pp. 103–111 (in Japanese).
安部恵介, 荒屋真二 (1986). 最長径路法を用いた列車運行シミュレーション. 情報処理学会論文誌, vol. 27, no. 1, pp. 103–111.

Acuna-Agost, R., Michelon, P., Feillet, D. and Gueye, S. (2011a). A MIP-based local search method for the railway rescheduling problem. *Networks*, vol. 57, no. 1, pp. 69–86. doi:10.1002/net.20384.

Acuna-Agost, R., Michelon, P., Feillet, D. and Gueye, S. (2011b). SAPI: Statistical analysis of propagation of incidents. A new approach for rescheduling trains after disruptions. *European Journal of Operational Research*, vol. 215, no. 1, pp. 227–243. doi:10.1016/j.ejor.2011.05.047.

Adenso-Díaz, B., González, M. O. and González-Torre, P. (1999). On-line timetable rescheduling in regional train services. *Transportation Research Part B: Methodological*, vol. 33, no. 6, pp. 387–398. doi:10.1016/S0191-2615(98)00041-1.

Afonso, P. A. and Bispo, C. F. (2011). Railway traffic management: Meet and pass problem. *Journal of System and Management Sciences*, vol. 1, no. 6, pp. 1–26.

Akita, K. and Hasegawa, Y. (2013). History of COMTRAC—Development of innovative traffic control system for Shinkansen. In *The Papers of Technical Meeting on "History of Electrical Engineering," IEE Japan*, Tokyo, Japan, no. HEE-13-9, pp. 1–6 (in Japanese). 秋田雄志, 長谷川豊 (2013). 新幹線輸送管理システム（コムトラック）開発の歴史. 電気学会研究会資料：電気技術史研究会, 東京, 日本, no. HEE-13-9, pp. 1–6.

Anderegg, L., Penna, P. and Widmayer, P. (2009). Online train disposition: To wait or not to wait? In Ahuja, R. K., Möhring, R. H. and Zaroliagis, C. D. eds., *Robust and Online Large-Scale Optimization*, Springer, Berlin, Germany, pp. 387–398. doi:10.1007/978-3-642-05465-5_17.

Araki, D., Sakurai, S., Ogawa, S. and Shinohara, M. (1995). Train schedule adjustment expert system based on cooperative man-machine interaction model. *IEEJ Transactions on Electronics, Information and Systems*, vol. 115, no. 5, pp. 672–680 (in Japanese). 荒木大, 櫻井茂明, 小川真一郎, 篠原正憲 (1995). マンマシン協調モデルに基づく運転整理エキスパートシステム. 電気学会論文誌C（電子・情報・システム部門誌）, vol. 115, no. 5, pp. 672–680.

Araya, S. and Abe, K. (1983). A heuristic control of train traffic in disturbed situations. *The Transactions of the Institute of Electrical Engineers of Japan C*, vol. 103, no. 11, pp. 249–256. doi:10.11526/ieejeiss1972.103.249 (in Japanese). 荒屋真二, 安部恵介 (1983). ヒューリスティクスを利用した列車運転整理手法. 電気学会論文誌 C, vol. 103, no. 11, pp. 249–256. doi:10.11526/ieejeiss1972.103.249.

Araya, S. and Sone, S. (1984). Traffic dynamics of automated transit systems with pre-established schedules. *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-14, no. 4, pp. 677–687. doi:10.1109/TSMC.1984.6313344.

Aronsson, M., Bohlin, M. and Kreuger, P. (2009). MILP formulations of cumulative constraints for railway scheduling—A comparative study. In Clausen, J. and Di Stefano, G. eds., *9th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS 2009)*, Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany. doi:10.4230/OASIcs.ATMOS.2009.2141.

Assad, A. A. (1980). Models for rail transportation. *Transportation Research Part A: General*, vol. 14, no. 3, pp. 205–220. doi:10.1016/0191-2607(80)90017-5.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P. and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, vol. 46, no. 3, pp. 316–329. doi:10.1287/opre.46.3.316.

Bauer, R. and Schöbel, A. (2014). Rules of thumb: Practical online-strategies for delay management. *Public Transport*, vol. 6, no. 1, pp. 85–105. doi:10.1007/s12469-013-0082-8.

Ben-Khedher, N., Kintanar, J., Queille, C. and Stripling, W. (1998). Schedule optimization at SNCF: From conception to day of departure. *Interfaces*, vol. 28, no. 1, pp. 6–23. doi:10.1287/inte.28.1.6.

Bender, M., Büttner, S. and Krumke, S. O. (2013). Online delay management on a single train line: Beyond competitive analysis. *Public Transport*, vol. 5, no. 3, pp. 243–266. doi:10.1007/s12469-013-0070-z.

Berger, A., Hoffmann, R., Lorenz, U. and Stiller, S. (2011). Online railway delay management: Hardness, simulation and computation. *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 87, no. 7, pp. 616–629. doi:10.1177/0037549710373571.

Berger, A., Blaar, C., Gebhardt, A., Müller-Hannemann, M. and Schnee, M. (2011). Passenger flow-oriented train disposition. In Demetrescu, C. and Halldórsson, M. M. eds., *Algorithms—ESA 2011*, Springer, Berlin, Germany, pp. 227–238. doi:10.1007/978-3-642-23719-5_20.

Biederbick, C. and Suhl, L. (2007). Decision support tools for customer-oriented dispatching. In Geraets, F., Kroon, L., Schoebel, A., Wagner, D. and Zaroliagis, C. D. eds., *Algorithmic Methods for Railway Optimization*, Springer, Berlin, Germany, pp. 171–183. doi:10.1007/978-3-540-74247-0_8.

Boccia, M., Mannino, C. and Vasilyev, I. (2013). The dispatching problem on multitrack territories: Heuristic approaches based on mixed integer linear programming. *Networks*, vol. 62, no. 4, pp. 315–326. doi:10.1002/net.21528.

Boland, N., Evans, I., Mears, C., Niven, T., Pattison, M., Wallace, M. and Waterer, H. (2012). Rail disruption: Passenger focused recovery. In Brebbia, C. A., Tomii, N., Mera, J. M., Ning, B. and Tzieropoulos, P. eds., *Computers in Railways XIII*, WIT Press, Southampton, the U.K., pp. 543–553. doi:10.2495/CR120461.

van den Boom, T. J. J. and De Schutter, B. (2006). Modelling and control of discrete event systems using switching max-plus-linear systems. *Control Engineering Practice*, vol. 14, no. 10, pp. 1199–1211. doi:10.1016/j.conengprac.2006.02.006.

van den Boom, T. J. J. and De Schutter, B. (2007). On a model predictive control algorithm for dynamic railway network management. In *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis (RailHannover2007)*, Hannover, Germany.

Borodin, A. and El-Yaniv, R. (2005). *Online Computation and Competitive Analysis*. Cambridge University Press, Cambridge, the U.K., Paperback edition.

Brouer, B. D., Dirksen, J., Pisinger, D., Plum, C. E. M. and Vaaben, B. (2013). The vessel schedule recovery problem (VSRP)—A MIP model for handling disruptions in liner shipping. *European Journal of Operational Research*, vol. 224, no. 2, pp. 362–374. doi:10.1016/j.ejor.2012.08.016.

Budai, G., Maróti, G., Dekker, R., Huisman, D. and Kroon, L. (2010). Rescheduling in passenger railways: The rolling stock rebalancing problem. *Journal of Scheduling*, vol. 13, no. 3, pp. 281–297. doi:10.1007/s10951-009-0133-9.

Bussieck, M. R., Winter, T. and Zimmermann, U. T. (1997). Discrete optimization in public rail transport. *Mathematical Programming*, vol. 79, no. 1, pp. 415–444. doi:10.1007/BF02614327.

Cacchiani, V. and Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research*, vol. 219, no. 3, pp. 727–737. doi:10.1016/j.ejor.2011.11.003.

Cacchiani, V., Caprara, A., Galli, L., Kroon, L., Maróti, G. and Toth, P. (2012). Railway rolling stock planning: Robustness against large disruptions. *Transportation Science*, vol. 46, no. 2, pp. 217–232. doi:10.1287/trsc.1110.0388.

Cadarso, L. and Marín, Á. (2011). Robust rolling stock in rapid transit networks. *Computers & Operations Research*, vol. 38, no. 8, pp. 1131–1142. doi:10.1016/j.cor.2010.10.029.

Cadarso, L. and Marín, Á. (2012). Integration of timetable planning and rolling stock in rapid transit networks. *Annals of Operations Research*, vol. 199, no. 1, pp. 113–135. doi:10.1007/s10479-011-0978-0.

Cadarso, L., Marín, Á. and Maróti, G. (2013). Recovery of disruptions in rapid transit networks. *Transportation Research Part E: Logistics and Transportation Review*, vol. 53, pp. 15–33. doi:10.1016/j.tre.2013.01.013.

Caimi, G., Fuchsberger, M., Laumanns, M. and Lüthi, M. (2012). A model predictive control approach for discrete-time rescheduling in complex central railway station areas. *Computers & Operations Research*, vol. 39, no. 11, pp. 2578–2593. doi:10.1016/j.cor.2012.01.003.

Caprara, A., Kroon, L., Monaci, M., Peeters, M. and Toth, P. (2007). Passenger railway optimization. In Barnhart, C. and Laporte, G. eds., *Transportation*, North-Holland, Amsterdam, the Netherlands, chap. 3, pp. 129–187. doi:10.1016/S0927-0507(06)14003-7.

Caprara, A., Galli, L., Kroon, L., Maróti, G. and Toth, P. (2010). Robust train routing and online re-scheduling. In Erlebach, T. and Lübbecke, M. eds., *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2010)*, Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 24–33. doi:10.4230/OASIcs.ATMOS.2010.24.

Caprara, A., Galli, L., Stiller, S. and Toth, P. (2014). Delay-robust event scheduling. *Operations Research*, vol. 62, no. 2, pp. 274–283. doi:10.1287/opre.2014.1259.

Chang, C. S. and Thia, B. S. (1996). Online rescheduling of mass rapid transit systems: Fuzzy expert system approach. *IEE Proceedings—Electric Power Applications*, vol. 143, no. 4, pp. 307–316. doi:10.1049/ip-epa:19960311.

Chang, S.-C. and Chung, Y.-C. (2005). From timetabling to train regulation—A new train operation model. *Information and Software Technology*, vol. 47, no. 9, pp. 575–585. doi:10.1016/j.infsof.2004.10.008.

Chen, L., Schmid, F., Dasigi, M., Ning, B., Roberts, C. and Tang, T. (2010). Real-time train rescheduling in junction areas. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 224, no. 6, pp. 547–557. doi:10.1243/09544097JRRT391.

Chigusa, K., Sato, K. and Koseki, T. (2012). Passenger-oriented optimization for train rescheduling on the basis of mixed integer programming. *IEEJ Transactions on Industry Applications*, vol. 132, no. 2, pp. 170–177. doi:10.1541/ieejias.132.170 (in Japanese).
千種健二, 佐藤圭介, 古関隆章 (2012). 混合整数計画法に基づく列車運行乱れ時の旅行時間増大量に主眼を置いた運転整理最適化. 電気学会論文誌D（産業応用部門誌）, vol. 132, no. 2, pp. 170–177. doi:10.1541/ieejias.132.170.

Cicerone, S., D'Angelo, G., Di Stefano, G., Frigioni, D. and Navarra, A. (2009a). Recoverable robustness for train shunting problems. *Algorithmic Operations Research*, vol. 4, no. 2, pp. 102–116.

Cicerone, S., D'Angelo, G., Di Stefano, G., Frigioni, D., Navarra, A., Schachtebeck, M. and Schöbel, A. (2009b). Recoverable robustness in shunting and timetabling. In Ahuja, R. K., Möhring, R. H. and Zaroliagis, C. D. eds., *Robust and Online Large-Scale Optimization*, Springer, Berlin, Germany, pp. 28–60. doi:10.1007/978-3-642-05465-5_2.

Cicerone, S., Di Stefano, G., Schachtebeck, M. and Schöbel, A. (2012). Multi-stage recovery robustness for optimization problems: A new concept for planning under disturbances. *Information Sciences*, vol. 190, pp. 107–126. doi:10.1016/j.ins.2011.12.010.

Clausen, J. (2007). Disruption management in passenger transportation—From air to tracks. In Liebchen, C., Ahuja, R. K. and Mesa, J. A. eds., *7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS 2007)*, Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 30–47. doi:10.4230/OASIcs.ATMOS.2007.1183.

Clausen, J., Larsen, A., Larsen, J. and Rezanova, N. J. (2010). Disruption management in the airline industry—Concepts, models and methods. *Computers & Operations Research*, vol. 37, no. 5, pp. 809–821. doi:10.1016/j.cor.2009.03.027.

Cordeau, J.-F., Toth, P. and Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science*, vol. 32, no. 4, pp. 380–404. doi:10.1287/trsc.32.4.380.

Corman, F., D'Ariano, A., Pacciarelli, D. and Pranzo, M. (2009a). Evaluation of green wave policy in real-time railway traffic management. *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 607–616. doi:10.1016/j.trc.2009.04.001.

Corman, F., Goverde, R. M. P. and D'Ariano, A. (2009b). Rescheduling dense train traffic over complex station interlocking areas. In Ahuja, R. K., Möhring, R. H. and Zaroliagis, C. D. eds., *Robust and Online Large-Scale Optimization*, Springer, Berlin, Germany, pp. 369–386. doi:10.1007/978-3-642-05465-5_16.

Corman, F., D'Ariano, A., Pacciarelli, D. and Pranzo, M. (2010a). Centralized versus distributed systems to reschedule trains in two dispatching areas. *Public Transport*, vol. 2, no. 3, pp. 219–247. doi:10.1007/s12469-010-0032-7.

Corman, F., D'Ariano, A., Pacciarelli, D. and Pranzo, M. (2010b). Railway dynamic traffic management in complex and densely used networks. In Negenborn, R. R., Lukszo, Z. and Hellendoorn, H. eds., *Intelligent Infrastructures*, Springer, Dordrecht, the Netherlands, chap. 15, pp. 377–404. doi:10.1007/978-90-481-3598-1_15.

Corman, F., D'Ariano, A., Pacciarelli, D. and Pranzo, M. (2010c). A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological*, vol. 44, no. 1, pp. 175–192. doi:10.1016/j.trb.2009.05.004.

Corman, F., D'Ariano, A., Pranzo, M. and Hansen, I. A. (2011a). Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. *Transportation Planning and Technology*, vol. 34, no. 4, pp. 341–362. doi:10.1080/03081060.2011.577152.

Corman, F., D'Ariano, A., Hansen, I. A. and Pacciarelli, D. (2011b). Optimal multi-class rescheduling of railway traffic. *Journal of Rail Transport Planning & Management*, vol. 1, no. 1, pp. 14–24. doi:10.1016/j.jrtpm.2011.06.001.

Corman, F. and D'Ariano, A. (2012). Assessment of advanced dispatching measures for recovering disrupted railway traffic situations. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2289, pp. 1–9. doi:10.3141/2289-01.

Corman, F., D'Ariano, A., Pacciarelli, D. and Pranzo, M. (2012a). Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 79–94. doi:10.1016/j.trc.2010.09.009.

Corman, F., D'Ariano, A., Pacciarelli, D. and Pranzo, M. (2012b). Optimal inter-area coordination of train rescheduling decisions. *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 71–88. doi:10.1016/j.tre.2011.05.002.

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

Crainic, T. G. and Laporte, G. (1997). Planning models for freight transportation. *European Journal of Operational Research*, vol. 97, no. 3, pp. 409–438. doi:10.1016/S0377-2217(96)00298-6.

D'Ariano, A., Pacciarelli, D. and Pranzo, M. (2007a). A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, vol. 183, no. 2, pp. 643–657. doi:10.1016/j.ejor.2006.10.034.

D'Ariano, A., Pranzo, M. and Hansen, I. A. (2007b). Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 208–222. doi:10.1109/TITS.2006.888605.

D'Ariano, A., Pacciarelli, D. and Pranzo, M. (2008a). Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 2, pp. 232–245. doi:10.1016/j.trc.2007.07.006.

D'Ariano, A., Corman, F., Pacciarelli, D. and Pranzo, M. (2008b). Reordering and local rerouting strategies to manage train traffic in real time. *Transportation Science*, vol. 42, no. 4, pp. 405–419. doi:10.1287/trsc.1080.0247.

D'Ariano, A. and Pranzo, M. (2009). An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, vol. 9, no. 1, pp. 63–84. doi:10.1007/s11067-008-9088-1.

De Almeida, D., Marcos, N., Vianey, S. and Weber, C. (2008). Robustness of rolling stock shedules. Technical Report ARRIVAL-TR-0129, Algorithms for Robust and online Railway optimization: Improving the Validity and reliAbility of Large scale systems (ARRIVAL).

De Giovanni, L., Heilporn, G. and Labbé, M. (2014). A polyhedral study for delay management in public transportation. *Procedia—Social and Behavioral Sciences*, vol. 108, pp. 15–25. doi:10.1016/j.sbspro.2013.12.816.

De Schutter, B., van den Boom, T. and Hegyi, A. (2002). Model predictive control approach for recovery from delays in railway systems. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1793, pp. 15–20. doi:10.3141/1793-03.

Demange, M., Di Stefano, G. and Leroy-Beaulieu, B. (2012). On the online track assignment problem. *Discrete Applied Mathematics*, vol. 160, no. 7–8, pp. 1072–1093. doi:10.1016/j.dam.2012.01.002.

Desrosiers, J. and Lübbecke, M. E. (2005). A primer in column generation. In Desaulniers, G., Desrosiers, J. and Solomon, M. M. eds., *Column Generation*, Springer, New York, the U.S., chap. 1, pp. 1–32. doi:10.1007/0-387-25486-2_1.

Dollevoet, T., Huisman, D., Schmidt, M. and Schöbel, A. (2012). Delay management with rerouting of passengers. *Transportation Science*, vol. 46, no. 1, pp. 74–89. doi:10.1287/trsc.1110.0375.

Dollevoet, T. and Huisman, D. (2014). Fast heuristics for delay management with passenger rerouting. *Public Transport*, vol. 6, no. 1, pp. 67–84. doi:10.1007/s12469-013-0076-6.

Dollevoet, T., Corman, F., D'Ariano, A. and Huisman, D. (2014). An iterative optimization framework for delay management and train scheduling. *Flexible Services and Manufacturing Journal*, vol. 26, no. 4, pp. 490–515. doi:10.1007/s10696-013-9187-2.

Dollevoet, T., Huisman, D., Kroon, L., Schmidt, M. and Schöbel, A. (2015). Delay management including capacities of stations. *Transportation Science*, vol. 49, no. 2, pp. 185–203. doi:10.1287/trsc.2013.0506.

Dündar, S. and Şahin, İ. (2013). Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways. *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 1–15. doi:10.1016/j.trc.2012.11.001.

Eberlein, X. J., Wilson, N. H. M. and Bernstein, D. (1999). Modeling real-time control strategies in public transit operations. In Wilson, N. H. M. ed., *Computer-Aided Transit Scheduling*, Springer, Berlin, Germany, pp. 325–346. doi:10.1007/978-3-642-85970-0_16.

Espinosa-Aranda, J. L. and García-Ródenas, R. (2013). A demand-based weighted train delay approach for rescheduling railway networks in real time. *Journal of Rail Transport Planning & Management*, vol. 3, no. 1–2, pp. 1–13. doi:10.1016/j.jrtpm.2013.10.001.

Fan, B., Roberts, C. and Weston, P. (2012). A comparison of algorithms for minimising delay costs in disturbed railway traffic scenarios. *Journal of Rail Transport Planning & Management*, vol. 2, no. 1–2, pp. 23–33. doi:10.1016/j.jrtpm.2012.09.002.

Fekete, S. P., Kröller, A., Lorek, M. and Pfetsch, M. (2011). Disruption management with rescheduling of trips and vehicle circulations. arXiv:1101.2123v1 [math.OC].

Fioole, P.-J., Kroon, L., Maróti, G. and Schrijver, A. (2006). A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, vol. 174, no. 2, pp. 1281–1297. doi:10.1016/j.ejor.2005.03.032.

Flamini, M. and Pacciarelli, D. (2008). Real time management of a metro rail terminus. *European Journal of Operational Research*, vol. 189, no. 3, pp. 746–761. doi:10.1016/j.ejor.2006.09.098.

Flier, H., Gaurav, A. and Nunkesser, M. (2007). Combinatorial aspects of move-up crews. Technical Report ARRIVAL-TR-0073, Algorithms for Robust and online Railway optimization: Improving the Validity and reliAbility of Large scale systems (ARRIVAL).

Flier, H., Nunkesser, M., Schachtebeck, M. and Schöbel, A. (2008). Integrating rolling stock circulation into the delay management problem. Technical Report ARRIVAL-TR-0132, Algorithms for Robust and online Railway optimization: Improving the Validity and reliAbility of Large scale systems (ARRIVAL).

Fujimori, A., Tomii, N. and Hirayama, J. (2004). A crew rescheduling algorithm based on combinatorial auction model. *Information Technology Letters*, vol. 3, pp. 15–18 (in Japanese).
藤森淳, 富井規雄, 平山純一郎 (2004). 組合せオークションモデルに基づく異常時の鉄道乗務員運用変更計画作成アルゴリズム. 情報科学技術レターズ, vol. 3, pp. 15–18.

Fukumura, N. and Maruyama, K. (2008). A station shunting rescheduling algorithm while railway traffic is disrupted. In *Abstracts of The 2008 Fall National Conference of Operations Research Society of Japan*, Sapporo, Japan, pp. 198–199 (in Japanese)†.
福村直登, 丸山公輔 (2008). 列車ダイヤ乱れ時の駅構内作業変更計画作成アルゴリズム. 日本オペレーションズ・リサーチ学会 2008 年秋季研究発表会アブストラクト集, 札幌, 日本, pp. 198–199.

Gatto, M., Glaus, B., Jacob, R., Peeters, L. and Widmayer, P. (2004). Railway delay management: Exploring its algorithmic complexity. In Hagerup, T. and Katajainen, J. eds., *Algorithm Theory—SWAT 2004*, Springer, Berlin, Germany, pp. 199–211. doi:10.1007/978-3-540-27810-8_18.

Gatto, M., Jacob, R., Peeters, L. and Schöbel, A. (2005). The computational complexity of delay management. In Kratsch, D. ed., *Graph-Theoretic Concepts in Computer Science*, Springer, Berlin, Germany, pp. 227–238. doi:10.1007/11604686_20.

Gatto, M., Jacob, R., Peeters, L. and Widmayer, P. (2007). Online delay management on a single train line. In Geraets, F., Kroon, L., Schoebel, A., Wagner, D. and Zaroliagis, C. D. eds., *Algorithmic Methods for Railway Optimization*, Springer, Berlin, Germany, pp. 306–320. doi:10.1007/978-3-540-74247-0_17.

Goodman, C. J. and Murata, S. (2001). Metro traffic regulation from the passenger perspective. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 215, no. 2, pp. 137–147. doi:10.1243/0954409011531468.

Goodman, C. J. and Takagi, R. (2004). Dynamic re-scheduling of trains after disruption. In Allan, J., Brebbia, C. A., Hill, R. J., Sciutto, G. and Sone, S. eds., *Computers in Railways IX*, WIT Press, Southampton, the U.K., pp. 765–774.

Goverde, R. M. P. (1998). Synchronization control of scheduled train services to minimize passenger waiting times. Institutional Repository, Delft University of Technology.

Gurobi Optimization (2016). Gurobi Optimizer. Retrieved January 3, 2016, from http://www.gurobi.com/products/gurobi-optimizer.

Hansen, I. A. and Pachl, J. eds. (2014). *Railway Timetabling & Operations*. EUrailpress, Hamburg, Germany, Second revised and extended edition.

Hara, K., Kojima, H., Henda, F. and Watanabe, T. (2004). Development of a total operation management system. *JR EAST Technical Review*, no. 3, pp. 63–72.

Harrod, S. S. (2012). A tutorial on fundamental model structures for railway timetable optimization. *Surveys in Operations Research and Management Science*, vol. 17, no. 2, pp. 85–96. doi:10.1016/j.sorms.2012.08.002.

Heilporn, G., De Giovanni, L. and Labbé, M. (2008). Optimization models for the single delay management problem in public transportation. *European Journal of Operational Research*, vol. 189, no. 3, pp. 762–774. doi:10.1016/j.ejor.2006.10.065.

Hillier, F. S. and Lieberman, G. J. (2014). *Introduction to Operations Research*. McGraw-Hill Education, New York, the U.S., Tenth edition.

Hirai, C., Tomii, N., Tashiro, Y., Kondou, S. and Fujimori, A. (2006). An algorithm for train rescheduling using rescheduling pattern description language R. In Allan, J., Brebbia, C. A., Rumsey, A. F., Sciutto, G., Sone, S. and Goodman, C. J. eds., *Computers in Railways X*, WIT Press, Southampton, the U.K., pp. 551–561. doi:10.2495/CR060551.

Hirai, C. and Takahashi, Y. (2009). Multi level approach in the feasibility check on the train stop deployment planning. In *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis (RailZurich2009)*, Zurich, Switzerland.

Hirai, C., Kunimatsu, T., Tomii, N., Kondou, S. and Takaba, M. (2009). A train stop deployment planning algorithm using a Petri-net-based modelling approach. *Quarterly Report of RTRI*, vol. 50, no. 1, pp. 8–13. `doi:`10.2219/rtriqr.50.8.

Hirai, C. (2010). Modelling analyses in train traffic control for rescheduling support systems. Doctoral dissertation, Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo, Japan.

Huisman, D., Kroon, L. G., Lentink, R. M. and Vromans, M. J. C. M. (2005). Operations research in passenger railway transportation. *Statistica Neerlandica*, vol. 59, no. 4, pp. 467–497. `doi:`10.1111/j.1467-9574.2005.00303.x.

Huisman, D. and Wagelmans, A. P. M. (2006). A solution approach for dynamic vehicle and crew scheduling. *European Journal of Operational Research*, vol. 172, no. 2, pp. 453–471. `doi:`10.1016/j.ejor.2004.10.009.

Huisman, D. (2007). A column generation approach for the rail crew re-scheduling problem. *European Journal of Operational Research*, vol. 180, no. 1, pp. 163–173. `doi:`10.1016/j.ejor.2006.04.026.

IBM (2016). IBM ILOG CPLEX Optimizer. Retrieved January 3, 2016, from http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

Inoue, K., Arisawa, T., Soma, S. and Henda, F. (2008). Development of operation rescheduling system. *JR EAST Technical Review*, no. 11, pp. 33–35.

Iqbal, S. M. Z., Grahn, H. and Törnquist Krasemann, J. (2013). Multi-strategy based train re-scheduling during railway traffic disturbances. In *Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis (RailCopenhagen2013)*, Lyngby, Denmark.

Ireland, P., Case, R., Fallis, J., Van Dyke, C., Kuehn, J. and Meketon, M. (2004). The Canadian Pacific Railway transforms operations by using models to develop its operating plans. *Interfaces*, vol. 34, no. 1, pp. 5–14. `doi:`10.1287/inte.1030.0055.

Japan Freight Railway Company (2006). Train Status. Retrieved July–November, 2006, from http://www.jrfreight.co.jp/i_daiya/index.html (in Japanese)†.
JR 貨物 (2006). 輸送情報. http://www.jrfreight.co.jp/i_daiya/index.html (アクセス期間: 2006 年 7 月–11 月).

Jespersen-Groth, J. and Clausen, J. (2006). Optimal reinsertion of cancelled train lines. IMM-Technical Report-2006-13, Technical University of Denmark.

Jespersen-Groth, J., Potthoff, D., Clausen, J., Huisman, D., Kroon, L., Maróti, G. and Nielsen, M. N. (2009a). Disruption management in passenger railway transportation. In Ahuja, R. K., Möhring, R. H. and Zaroliagis, C. D. eds., *Robust and Online Large-Scale Optimization*, Springer, Berlin, Germany, pp. 399–421. doi:10.1007/978-3-642-05465-5_18.

Jespersen-Groth, J., Clausen, J. and Larsen, J. (2009b). Rolling stock recovery problem. IMM-Technical Report-2008-18, Technical University of Denmark.

Kamada, T. (2009). Blank Map with Prefectures Ver8.4. Retrieved October 26, 2009, from http://www5b.biglobe.ne.jp/~t-kamada/CBuilder/kenmap.htm (in Japanese)†.
鎌田輝男 (2009). 白地図 KenMap Ver8.4. http://www5b.biglobe.ne.jp/~t-kamada/CBuilder/kenmap.htm (アクセス日: 2009 年 10 月 26 日).

Kanai, S., Shiina, K., Harada, S. and Tomii, N. (2011). An optimal delay management algorithm from passengers' viewpoints considering the whole railway network. *Journal of Rail Transport Planning & Management*, vol. 1, no. 1, pp. 25–37. doi:10.1016/j.jrtpm.2011.09.003.

Kecman, P., Corman, F., D'Ariano, A. and Goverde, R. M. P. (2013). Rescheduling models for railway traffic management in large-scale networks. *Public Transport*, vol. 5, no. 1, pp. 95–123. doi:10.1007/s12469-013-0063-y.

Kersbergen, B., van den Boom, T. and De Schutter, B. (2013). On implicit versus explicit max-plus modeling for the rescheduling of trains. In *Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis (RailCopenhagen2013)*, Lyngby, Denmark.

Khosravi, B., Bennell, J. A. and Potts, C. N. (2012). Train scheduling and rescheduling in the UK with a modified shifting bottleneck procedure. In Delling, D. and Liberti, L. eds., *12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2012)*, Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 120–131. doi:10.4230/OASIcs.ATMOS.2012.120.

Kitahara, F., Iwamoto, T., Ito, S., Fujiwara, K. and Fujiwara, M. (1998). Development of autonomous decentralized transport operation control system for super high-density railway lines. *IEEJ Transactions on Industry Applications*, vol. 118, no. 4, pp. 534–541. doi:10.1541/ieejias.118.534 (in Japanese).
北原文夫, 岩本孝雄, 伊藤聡, 藤原和紀, 藤原道雄 (1998). 超高密度鉄道の列車群を自律分散制御する東京圏輸送管理システムの開発. 電気学会論文誌Ｄ（産業応用部門誌）, vol. 118, no. 4, pp. 534–541. doi:10.1541/ieejias.118.534.

Kliewer, N. and Suhl, L. (2011). A note on the online nature of the railway delay management problem. *Networks*, vol. 57, no. 1, pp. 28–37. doi:10.1002/net.20381.

Kojima, T., Asami, M., Soma, S. and Henda, F. (2008). Field test of crew operation rescheduling support system. *JR EAST Technical Review*, no. 11, pp. 28–32.

Komaya, K. and Fukuda, T. (1991). A knowledge-based approach for railway scheduling. In *Proceedings of the 7th IEEE Conference on Artificial Intelligence Applications (CAIA '91)*, Miami Beach, the U.S., pp. 404–411. doi:10.1109/CAIA.1991.120901.

Kroon, L., Huisman, D., Abbink, E., Fioole, P.-J., Fischetti, M., Maróti, G., Schrijver, A., Steenbeek, A. and Ybema, R. (2009). The new Dutch timetable: The OR revolution. *Interfaces*, vol. 39, no. 1, pp. 6–17. doi:10.1287/inte.1080.0409.

Kroon, L. and Huisman, D. (2011). Algorithmic support for railway disruption management. In van Nunen, J. A. E. E., Huijbregts, P. and Rietveld, P. eds., *Transitions Towards Sustainable Mobility*, Springer, Berlin, Germany, chap. 11, pp. 193–210. doi:10.1007/978-3-642-21192-8_11.

Kroon, L., Maróti, G. and Nielsen, L. (2015). Rescheduling of railway rolling stock with dynamic passenger flows. *Transportation Science*, vol. 49, no. 2, pp. 165–184. doi:10.1287/trsc.2013.0502.

Krumke, S. O., Thielen, C. and Zeck, C. (2011). Extensions to online delay management on a single train line: New bounds for delay minimization and profit maximization. *Mathematical Methods of Operations Research*, vol. 74, no. 1, pp. 53–75. doi:10.1007/s00186-011-0349-2.

Kumazawa, K., Hara, K. and Koseki, T. (2008). A novel train rescheduling algorithm for correcting disrupted train operations in a dense urban environment. In Allan, J., Arias, E., Brebbia, C. A., Goodman, C. J., Rumsey, A. F., Sciutto, G. and Tomii, N. eds., *Computers in Railways XI*, WIT Press, Southampton, the U.K., pp. 565–574. doi:10.2495/CR080551.

Kunimatsu, T. and Hirai, C. (2009). A train rescheduling algorithm using estimated passenger demand. *RTRI Report*, vol. 23, no. 8, pp. 17–22 (in Japanese).
國松武俊, 平井力 (2009). 利用者デマンドを考慮した運転整理案作成アルゴリズムの開発. 鉄道総研報告, vol. 23, no. 8, pp. 17–22.

Kunimatsu, T. and Hirai, C. (2014). The prediction of passenger flow under transport disturbance using accumulated passenger data. In Brebbia, C. A., Tomii, N., Tzieropoulos, P. and Mera, J. M. eds., *Computers in Railways XIV*, WIT Press, Southampton, the U.K., pp. 623–634. doi:10.2495/CR140521.

Kwan, C. M. and Chang, C. S. (2005). Application of evolutionary algorithm on a transportation scheduling problem—The mass rapid transit. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, the U.K., vol. 2, pp. 987–994. doi:10.1109/CEC.2005.1554798.

Lai, Y.-C., Fan, D.-C. and Huang, K.-L. (2015). Optimizing rolling stock assignment and maintenance plan for passenger railway operations. *Computers & Industrial Engineering*, vol. 85, pp. 284–295. doi:10.1016/j.cie.2015.03.016.

Lamorgese, L. and Mannino, C. (2015). An exact decomposition approach for the real-time train dispatching problem. *Operations Research*, vol. 63, no. 1, pp. 48–64. doi:10.1287/opre.2014.1327.

Li, C., Qi, X. and Lee, C.-Y. (2015). Disruption recovery for a vessel in liner shipping. *Transportation Science*, vol. 49, no. 4, pp. 900–921. doi:10.1287/trsc.2015.0589.

Liebchen, C., Lübbecke, M., Möhring, R. and Stiller, S. (2009). The concept of recoverable robustness, linear programming recovery, and railway applications. In Ahuja, R. K., Möhring, R. H. and Zaroliagis, C. D. eds., *Robust and Online Large-Scale Optimization*, Springer, Berlin, Germany, pp. 1–27. doi:10.1007/978-3-642-05465-5_1.

Liu, S. Q. and Kozan, E. (2009). Scheduling trains as a blocking parallel-machine job shop scheduling problem. *Computers & Operations Research*, vol. 36, no. 10, pp. 2840–2852. doi:10.1016/j.cor.2008.12.012.

Louwerse, I. (2009). Adjusting the rolling stock allocation in case of disruptions. Bachelor's thesis, Erasmus School of Economics, Erasmus University Rotterdam, Rotterdam, the Netherlands.

Louwerse, I. and Huisman, D. (2014). Adjusting a railway timetable in case of partial or complete blockades. *European Journal of Operational Research*, vol. 235, no. 3, pp. 583–593. doi:10.1016/j.ejor.2013.12.020.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, vol. 53, no. 6, pp. 1007–1023. doi:10.1287/opre.1050.0234.

Luethi, M., Medeossi, G. and Nash, A. (2009). Structure and simulation evaluation of an integrated real-time rescheduling system for railway networks. *Networks and Spatial Economics*, vol. 9, no. 1, pp. 103–121. doi:10.1007/s11067-008-9085-4.

Lusby, R. M., Larsen, J., Ehrgott, M. and Ryan, D. M. (2013). A set packing inspired method for real-time junction train routing. *Computers & Operations Research*, vol. 40, no. 3, pp. 713–724. doi:10.1016/j.cor.2011.12.004.

Mannino, C. and Mascis, A. (2009). Optimal real-time traffic control in metro stations. *Operations Research*, vol. 57, no. 4, pp. 1026–1039. doi:10.1287/opre.1080.0642.

Maróti, G. and Kroon, L. (2005). Maintenance routing for train units: The transition model. *Transportation Science*, vol. 39, no. 4, pp. 518–525. doi:10.1287/trsc.1050.0116.

Maróti, G. and Kroon, L. (2007). Maintenance routing for train units: The interchange model. *Computers & Operations Research*, vol. 34, no. 4, pp. 1121–1140. doi:10.1016/j.cor.2005.05.026.

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

Mascis, A., Pacciarelli, D. and Pranzo, M. (2008). Scheduling models for short-term railway traffic optimisation. In Hickman, M., Mirchandani, P. and Voß, S. eds., *Computer-aided Systems in Public Transport*, Springer, Berlin, Germany, pp. 71–90. doi:10.1007/978-3-540-73312-6_5.

Mazzarello, M. and Ottaviani, E. (2007). A traffic management system for real-time traffic optimisation in railways. *Transportation Research Part B: Methodological*, vol. 41, no. 2, pp. 246–274. doi:10.1016/j.trb.2006.02.005.

Meng, L. and Zhou, X. (2011). Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. *Transportation Research Part B: Methodological*, vol. 45, no. 7, pp. 1080–1102. doi:10.1016/j.trb.2011.05.001.

Meng, X., Jia, L. and Qin, Y. (2010). Train timetable optimizing and rescheduling based on improved particle swarm algorithm. *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2197, pp. 71–79. doi:10.3141/2197-09.

Min, Y.-H., Park, M.-J., Hong, S.-P. and Hong, S.-H. (2011). An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network. *Transportation Research Part B: Methodological*, vol. 45, no. 2, pp. 409–429. doi:10.1016/j.trb.2010.08.001.

Ministry of Land, Infrastructure, Transport and Tourism (2012). Public Transport Policy: The 11th Metropolis Transport Census Investigation Result Spreadsheets. Retrieved December 2, 2012, from http://www.mlit.go.jp/sogoseisaku/transport/sosei_transport_tk_000035.html (in Japanese)[†].
国土交通省 (2012). 公共交通政策：第１１回大都市交通センサス調査結果集計表. http://www.mlit.go.jp/sogoseisaku/transport/sosei_transport_tk_000035.html (アクセス日: 2012 年 12 月 2 日).

Mladenović, S. and Čangalović, M. (2007). Heuristic approach to train rescheduling. *Yugoslav Journal of Operations Research*, vol. 17, no. 1, pp. 9–29.

Nagai, T., Ebihara, K., Tokairin, T. and Nakata, H. (1977). Evaluation of train operation adjustment. In *Presented Papers of the 14th Symposium on the Use of Cybernetics on the Railway*, Tokyo, Japan, pp. 220–224 (in Japanese).
長井忠昌, 海老原浩一, 東海林保, 中田裕友 (1977). 運転整理の評価. 第 14 回鉄道におけるサイバネティクス利用国内シンポジウム論文集, 東京, 日本, pp. 220–224.

Nagasaki, Y., Eguchi, M. and Koseki, T. (2003). Automatic generation and evaluation of urban railway rescheduling plan. In *Proceedings of International Symposium on Speed-up and Service Technology for Railway and Maglev Systems 2003 (STECH'03)*, Tokyo, Japan, pp. 301–306.

Nakamura, T., Hirai, C. and Nishioka, Y. (2011). A practical train rescheduling algorithm using three predetermined factors. In *Proceedings of the 4th International Seminar on Railway Operations Modelling and Analysis (RailRome2011)*, Rome, Italy.

Narayanaswami, S. and Rangaraj, N. (2011). Scheduling and rescheduling of railway operations: A review and expository analysis. *Technology Operation Management*, vol. 2, no. 2, pp. 102–122. doi:10.1007/s13727-012-0006-x.

Nemhauser, G. L. and Wolsey, L. A. (1999). *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, the U.S.

Nielsen, L. K. (2011). Rolling stock rescheduling in passenger railways. Ph.D. thesis, Erasmus Research Institute of Management, Erasmus University Rotterdam, Rotterdam, the Netherlands.

Nielsen, L. K., Kroon, L. and Maróti, G. (2012). A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, vol. 220, no. 2, pp. 496–509. doi:10.1016/j.ejor.2012.01.037.

O'Dell, S. W. and Wilson, N. H. M. (1999). Optimal real-time control strategies for rail transit operations during disruptions. In Wilson, N. H. M. ed., *Computer-Aided Transit Scheduling*, Springer, Berlin, Germany, pp. 299–323. doi:10.1007/978-3-642-85970-0_15.

ON-TIME (2013). Task 3.1 Assessment of state-of-art of train timetabling. Internal Report ONT-WP03-I-EPF-008-03, Optimal Networks for Train Integration Management across Europe (ON-TIME).

Otsuki, T., Aisu, H. and Tanaka, T. (2010). The solution for constraint satisfaction problems of railway rolling stock allocation. *Transactions of the Operations Research Society of Japan*, vol. 53, pp. 30–55 (in Japanese).
大槻知史, 愛須英之, 田中俊明 (2010). 車両運用計画問題に対する制約充足解法の提案. 日本オペレーションズ・リサーチ学会和文論文誌, vol. 53, pp. 30–55.

Pellegrini, P., Marlière, G. and Rodriguez, J. (2014). Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Research Part B: Methodological*, vol. 59, pp. 58–80. doi:10.1016/j.trb.2013.10.013.

Planning Systems Laboratory, Railway Technical Research Institute (2005). *Scheduling Algorithms in Railways*. NTS, Tokyo, Japan (in Japanese).
（財）鉄道総合技術研究所運転システム研究室 (2005). 鉄道のスケジューリングアルゴリズム. エヌ・ティー・エス, 東京, 日本.

Potthoff, D., Huisman, D. and Desaulniers, G. (2010). Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science*, vol. 44, no. 4, pp. 493–505. doi:10.1287/trsc.1100.0322.

Qi, X. (2015). Disruption management for liner shipping. In Lee, C.-Y. and Meng, Q. eds., *Handbook of Ocean Container Transport Logistics*, Springer, Cham, Switzerland, chap. 8, pp. 231–249. doi:10.1007/978-3-319-11891-8_8.

Railway Bureau, Ministry of Land, Infrastructure, Transport and Tourism (2012). *An Evaluation Method Manual of Rail Projects: 2012 Revised Edition*. Institution for Transport Policy Studies, Tokyo, Japan (in Japanese)†.
国土交通省鉄道局（監修） (2012). 鉄道プロジェクトの評価手法マニュアル 2012 年改訂版. 運輸政策研究機構, 東京, 日本.

Railway Freight Association (2006). *2006 Japan Freight Railway Company Timetable*. Railway Freight Association, Tokyo, Japan (in Japanese)†.
鉄道貨物協会 (2006). 2006 JR 貨物時刻表. 鉄道貨物協会, 東京, 日本.

Rezanova, N. J. and Ryan, D. M. (2010). The train driver recovery problem—A set partitioning based model and solution method. *Computers & Operations Research*, vol. 37, no. 5, pp. 845–856. doi:10.1016/j.cor.2009.03.023.

Rodriguez, J. (2007). A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B: Methodological*, vol. 41, no. 2, pp. 231–245. doi:10.1016/j.trb.2006.02.006.

Sakikawa, S., Sato, T., Morita, T. and Ohta, K. (2010). A method of train regulation planning by the constraint programming using train regulation knowledge. *IEEJ Transactions on Electronics, Information and Systems*, vol. 130, no. 2, pp. 332–342. doi:10.1541/ieejeiss.130.332 (in Japanese).
崎川修一郎 and 佐藤達広 and 森田豊久 and 大田健二 (2010). 運転整理知識を活用した制約プログラミングによる整理案作成方式. 電気学会論文誌 C（電子・情報・システム部門誌）, vol. 130, no. 2, pp. 332–342. doi:10.1541/ieejeiss.130.332.

Sato, T., Kakumoto, Y. and Murata, T. (2007). Shunting scheduling method in a railway depot for dealing with changes in operational conditions. *IEEJ Transactions on Electronics, Information and Systems*, vol. 127, no. 2, pp. 274–283. doi:10.1541/ieejeiss.127.274 (in Japanese).
佐藤達弘, 角本喜紀, 村田智洋 (2007). 条件変化に伴う再計画を考慮した鉄道車両基地構内入換スケジューリング方式. 電気学会論文誌 C（電子・情報・システム部門誌）, vol. 127, no. 2, pp. 274–283. doi:10.1541/ieejeiss.127.274.

Sato, T., Sakikawa, S., Morita, T., Ueki, N. and Murata, T. (2009). Crew and vehicle rescheduling based on a network flow model and its application to a railway train operation. *IAENG International Journal of Applied Mathematics*, vol. 39, no. 3, pp. 142–150.

Sato, T., Tomiyama, T., Morita, T. and Murata, T. (2012). A Lagrangian relaxation method for crew and vehicle rescheduling of railway passenger transportation and its application. *IEEJ Transactions on Electronics, Information and Systems*, vol. 132, no. 2, pp. 260–268. doi:10.1541/ieejeiss.132.260.

Schaafsma, A. A. M. and Bartholomeus, M. M. G. P. (2007). Dynamic traffic management in the Schiphol bottleneck. In *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis (RailHannover2007)*, Hannover, Germany.

Schachtebeck, M. and Schöbel, A. (2010). To wait or not to wait—And who goes first? Delay management with priority decisions. *Transportation Science*, vol. 44, no. 3, pp. 307–321. doi:10.1287/trsc.1100.0318.

Schmidt, M. (2013). Simultaneous optimization of delay management decisions and passenger routes. *Public Transport*, vol. 5, no. 1, pp. 125–147. doi:10.1007/s12469-013-0069-5.

Schöbel, A. (2001). A model for the delay management problem based on mixed-integer-programming. *Electronic Notes in Theoretical Computer Science*, vol. 50, no. 1, pp. 1–10. doi:10.1016/S1571-0661(04)00160-4.

Schöbel, A. (2006). *Optimization in Public Transportation*. Springer, New York, the U.S.

Schöbel, A. (2009). Capacity constraints in delay management. *Public Transport*, vol. 1, no. 2, pp. 135–154. doi:10.1007/s12469-009-0010-0.

Shen, S. and Wilson, N. H. M. (2001). An optimal integrated real-time disruption control model for rail transit systems. In Voß, S. and Daduna, J. R. eds., *Computer-Aided Scheduling of Public Transport*, Springer, Berlin, Germany, pp. 335–363. doi:10.1007/978-3-642-56423-9_19.

Shibghatullah, A. S. (2008). A multi-agent system for a bus crew rescheduling system. Ph.D. thesis, School of Information Systems, Computing and Mathematics, Brunel University London, London, the U.K.

Shimizu, H., Fujimoto, H., Takahashi, J. and Nozue, N. (2001). Constraints based effective railway rescheduling system. In *Proceedings of the 5th World Congress on Railway Research (WCRR 2001)*, Cologne, Germany.

Shimizu, H., Tanabe, H. and Yamamoto, M. (2008). The proposal system for Shinkansen using constraint programming. In *Proceedings of the 8th World Congress on Railway Research (WCRR 2008)*, Seoul, the Republic of Korea.

Sugi, M., Nagai, H., Yamamoto, M., Shiomi, Y., and Ota, J. (2010). Rescheduling of train shunting in railway stations. *International Journal of Automation Technology*, vol. 4, no. 6, pp. 495–501.

Suhl, L. and Mellouli, T. (1999). Requirement for, and design of, an operations control system for railways. In Wilson, N. H. M. ed., *Computer-Aided Transit Scheduling*, Springer, Berlin, Germany, pp. 371–390. doi:10.1007/978-3-642-85970-0_18.

Suhl, L., Biederbick, C. and Kliewer, N. (2001a). Design of customer-oriented dispatching support for railways. In Voß, S. and Daduna, J. R. eds., *Computer-Aided Scheduling of Public Transport*, Springer, Berlin, Germany, pp. 365–386. doi:10.1007/978-3-642-56423-9_20.

Suhl, L., Mellouli, T., Biederbick, C. and Goecke, J. (2001b). Managing and preventing delays in railway traffic by simulation and optimization. In Pursula, M. and Niittymäki, J. eds., *Mathematical Methods on Optimization in Transportation Systems*, Springer, New York, the U.S., chap. 1, pp. 3–16. doi:10.1007/978-1-4757-3357-0_1.

Takagi, R., Goodman, C. J. and Roberts, C. (2004). Optimising departure times at a transport interchange to improve connections when services are disrupted. In Allan, J., Brebbia, C. A., Hill, R. J., Sciutto, G. and Sone, S. eds., *Computers in Railways IX*, WIT Press, Southampton, the U.K., pp. 707–714.

Takahashi, S., Kataoka, K., Kojima, T. and Asami, M. (2008). An algorithm for automatically modifying train crew schedule. *IEEJ Transactions on Industry Applications*, vol. 128, no. 11, pp. 1291–1297. doi:10.1541/ieejias.128.1291 (in Japanese).
高橋理, 片岡健司, 小島央士, 浅見雅之 (2008). ダイヤ乱れ時における列車乗務員運用整理案の自動作成. 電気学会論文誌Ｄ（産業応用部門誌）, vol. 128, no. 11, pp. 1291–1297. doi:10.1541/ieejias.128.1291.

Technical Research Committee on Advanced Rail Transport Planning and Management, The Institute of Electrical Engineers of Japan ed. (2010). *Techniques for Restoring Disrupted Railway Traffic*. Ohmsha, Tokyo, Japan (in Japanese)†.
電気学会・鉄道における運行計画・運行管理業務高度化に関する調査専門委員会 (2010). 鉄道ダイヤ回復の技術. オーム社, 東京, 日本.

Tomii, N. and Ikeda, H. (1995). A hybrid train traffic rescheduling simulator. *Quarterly Report of RTRI*, vol. 36, no. 4, pp. 192–197.

Tomii, N., Tashiro, Y., Tanabe, N., Hirai, C. and Muraki, K. (2005). Train rescheduling algorithm which minimizes passengers' dissatisfaction. In Ali, M. and Esposito, F. eds., *Innovations in Applied Artificial Intelligence*, Springer, Berlin, Germany, pp. 829–838. doi:10.1007/11504894_113.

Tomiyama, T., Sato, T., Iwamura, S., Sakamoto, O. and Matsuda, K. (2012a). Development of rescheduling algorithm using constraint programming for train unit operation with storage constraint. In *Proceedings of the 2012 IEE-Japan Industry Applications Society Conference (JIASC2012)*, Tsudanuma, Japan, vol. 5, pp. 267–270 (in Japanese).
富山友恵, 佐藤達広, 岩村篤樹, 坂本調, 松田和則 (2012a). 制約プログラミングを用いた滞泊制約付き車両運用整理手法の開発. 平成 24 年電気学会産業応用部門大会 (JIASC2012), 津田沼, 日本, vol. 5, pp. 267–270.

Tomiyama, T., Sato, T., Okada, K. and Wakamiya, T. (2012b). Rescheduling algorithm using operational patterns for rolling stock operation at train depots. In Brebbia, C. A., Tomii, N., Mera, J. M., Ning, B. and Tzieropoulos, P. eds., *Computers in Railways XIII*, WIT Press, Southampton, the U.K., pp. 555–566. doi:10.2495/CR120471.

Törnquist, J. and Persson, J. A. (2005). Train traffic deviation handling using tabu search and simulated annealing. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05)*, the Island of Hawaii, the U.S., no. 73a. doi:10.1109/HICSS.2005.641.

Törnquist, J. (2006). Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms. In Kroon, L. G. and Möhring, R. H. eds., *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2005)*, Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany. doi:10.4230/OASIcs.ATMOS.2005.659.

Törnquist, J. (2007). Railway traffic disturbance management—An experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. *Transportation Research Part A: Policy and Practice*, vol. 41, no. 3, pp. 249–266. doi:10.1016/j.tra.2006.05.003.

Törnquist, J. and Persson, J. A. (2007). *N*-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B: Methodological*, vol. 41, no. 3, pp. 342–362. doi:10.1016/j.trb.2006.06.001.

Törnquist Krasemann, J. (2012). Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. *Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 62–78. doi:10.1016/j.trc.2010.12.004.

Tsuchiya, R., Sugiyama, Y., Yamauchi, K., Fujinami, K., Arisawa, R. and Nakagawa, T. (2006). Route-choice support system for passengers in the face of unexpected disturbance of train operations. In Allan, J., Brebbia, C. A., Rumsey, A. F., Sciutto, G., Sone, S. and Goodman, C. J. eds., *Computers in Railways X*, WIT Press, Southampton, the U.K., pp. 189–197. doi:10.2495/CR060191.

Umetani, S. and Yagiura, M. (2007). Relaxation heuristics for the set covering problem. *Journal of the Operations Research Society of Japan*, vol. 50, no. 4, pp. 350–375.

Vanderbeck, F. (2005). Implementing mixed integer column generation. In Desaulniers, G., Desrosiers, J. and Solomon, M. M. eds., *Column Generation*, Springer, New York, the U.S., chap. 12, pp. 331–358. doi:10.1007/0-387-25486-2_12.

Veelenturf, L. P., Nielsen, L. K., Maróti, G. and Kroon, L. G. (2011). Passenger oriented disruption management by adapting stopping patterns and rolling stock schedules. In *Proceedings of the 4th International Seminar on Railway Operations Modelling and Analysis (RailRome2011)*, Rome, Italy.

Visentini, M. S., Borenstein, D., Li, J.-Q. and Mirchandani, P. B. (2014). Review of real-time vehicle schedule recovery methods in transportation services. *Journal of Scheduling*, vol. 17, no. 6, pp. 541–567. doi:10.1007/s10951-013-0339-8.

de Vries, R., De Schutter, B. and De Moor, B. (1998). On max-algebraic models for transportation networks. In *Proceedings of the 4th International Workshop on Discrete Event Systems (WODES'98)*, Cagliari, Italy, pp. 457–462.

Walker, C. G., Snowdon, J. N. and Ryan, D. M. (2005). Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers & Operations Research*, vol. 32, no. 8, pp. 2077–2094. doi:10.1016/j.cor.2004.02.001.

Wegele, S. and Schnieder, E. (2005). Dispatching of train operations using genetic algorithms. In *Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis (RailDelft2005)*, Delft, the Netherlands.

Williams, H. P. (2013). *Model Building in Mathematical Programming*. John Wiley & Sons, Chichester, the U.K., Fifth edition.

Winter, T. and Zimmermann, U. T. (2000). Real-time dispatch of trams in storage yards. *Annals of Operations Research*, vol. 96, no. 1, pp. 287–315. doi:10.1023/A:1018907720194.

Wolsey, L. A. (1998). *Integer Programming*. John Wiley & Sons, New York, the U.S.

Yang, Z., Koseki, T. and Sone, S. (2002). Automatic generation of rescheduling plan immediately after resumption of train operation using selective stops schedule pattern. *IEEJ Transactions on Industry Applications*, vol. 122, no. 2, pp. 104–110. doi:10.1541/ieejias.122.104 (in Japanese).
楊中平, 古関隆章, 曽根悟 (2002). 選択停車を利用した運転再開直後の運転整理案の自動生成. 電気学会論文誌Ｄ（産業応用部門誌）, vol. 122, no. 2, pp. 104–110. doi:10.1541/ieejias.122.104.

Yasumoto, T., Otsuki, T., Nagayoshi, S. and Nakajima, M. (2009). Development of a rolling stock scheduling system—Coping with disrupted traffic based on arrival time prediction. In *The Collection of Technical Papers on the 46th Railway Cybernetics and Symposium*, Osaka, Japan, Paper ID: 540 (in Japanese)†.
安本高典, 大槻知史, 永吉申作, 中島正貴 (2009). 車両運用計画システムの開発—到着予想に基づくダイヤ乱れへの対応—. 第 46 回鉄道サイバネ・シンポジウム論文集, 大阪, 日本, 論文番号 540.

† The references with this symbol have no official English title etc. and the translation is done by the author of this thesis.

Publications by the Author

Peer-reviewed Journal Articles

Sato, K., Tamura, K. and Tomii, N. (2013). A MIP-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. *Journal of Rail Transport Planning & Management*, vol. 3, no. 3, pp. 38–53. doi:10.1016/j.jrtpm.2013.10.007.

Sato, K. and Fukumura, N. (2012). Real-time freight locomotive rescheduling and uncovered train detection during disruption. *European Journal of Operational Research*, vol. 221, no. 3, pp. 636–648. doi:10.1016/j.ejor.2012.04.025.

Sato, K. and Fukumura, N. (2011a). Real-time freight train driver rescheduling during disruption. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E94-A, no. 6, pp. 1222–1229. doi:10.1587/transfun.E94.A.1222.

Peer-reviewed Conference Papers

Tamura, K., Tomii, N. and Sato, K. (2013a). An optimal rescheduling algorithm from passengers' viewpoint based on mixed integer programming formulation. In *Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis (RailCopenhagen2013)*, Lyngby, Denmark.

Sato, K. and Fukumura, N. (2010a). Freight locomotive rescheduling and uncovered train detection during disruptions. In *Selected Proceedings of the 12th World Conference on Transport Research (12th WCTR)*, Lisbon, Portugal, ID: 02693.

Non-peer-reviewed Technical Reports

Sakaguchi, T. and Sato, K. (2011). Rolling stock rescheduling algorithm for passenger trains during disruption of train operation. *RTRI Report*, vol. 25, no. 12, pp. 17–22 (in Japanese).
坂口隆, 佐藤圭介 (2011). 旅客列車を対象とした車両運用整理アルゴリズムの開発. 鉄道総研報告, vol. 25, no. 12, pp. 17–22.

Sato, K. and Fukumura, N. (2011b). Freight locomotive rescheduling algorithm under disordered train operation. *Quarterly Report of RTRI*, vol. 52, no. 2, pp. 81–85. doi:10.2219/rtriqr.52.81.

Sato, K. and Fukumura, N. (2010b). Freight locomotive rescheduling algorithm under disordered train operation. *RTRI Report*, vol. 24, no. 10, pp. 17–22 (in Japanese).
佐藤圭介, 福村直登 (2010b). 機関車運用整理案作成アルゴリズムの開発. 鉄道総研報告, vol. 24, no. 10, pp. 17–22.

Sato, K. and Fukumura, N. (2010c). An algorithm for freight train driver rescheduling in disruption situations. *Quarterly Report of RTRI*, vol. 51, no. 2, pp. 72–76. doi:10.2219/rtriqr.51.72.

Sato, K. and Fukumura, N. (2009a). Freight train driver rescheduling algorithm after disruptions. *RTRI Report*, vol. 23, no. 8, pp. 23–28 (in Japanese).
佐藤圭介, 福村直登 (2009a). 貨物列車の乗務員運用整理案作成アルゴリズムの開発. 鉄道総研報告, vol. 23, no. 8, pp. 23–28.

Non-peer-reviewed Magazine Articles

Sato, K. (2015a). A mixed integer programming formulation of timetable rescheduling minimizing further inconvenience to passengers. *Communications of the Operations Research Society of Japan*, vol. 60, no. 10, pp. 622–629 (in Japanese).
佐藤圭介 (2015a). 旅客が被る不便の増分を最小化する運転整理の定式化. オペレーションズ・リサーチ：経営の科学, vol. 60, no. 10, pp. 622–629.

Sato, K. (2014). Swift rolling stock rescheduling while traffic is disrupted. *RRR*, vol. 71, no. 1, pp. 16–19 (in Japanese).
佐藤圭介 (2014). ダイヤ乱れに応じて車両運用を迅速に変更する. RRR, vol. 71, no. 1, pp. 16–19.

Tomii, N. and Sato, K. (2013a). Recent trends of ICT application to railway operation and signaling systems: Train rescheduling in railways—State of the arts and the future—. *IPSJ Magazine*, vol. 54, no. 12, pp. 1254–1261 (in Japanese).
富井規雄, 佐藤圭介 (2013a). 鉄道の運行システムにおける情報処理技術の動向：鉄道のダイヤ乱れ時への対応＜その 2 ＞—現状と研究開発の状況—. 情報処理, vol. 54, no. 12, pp. 1254–1261.

Tomii, N. and Sato, K. (2013b). Recent trends of ICT application to railway operation and signaling systems: Train rescheduling in railways—State of the arts and the future—. *IPSJ Magazine*, vol. 54, no. 11, pp. 1164–1170 (in Japanese).
富井規雄, 佐藤圭介 (2013b). 鉄道の運行システムにおける情報処理技術の動向：鉄道のダイヤ乱れ時への対応＜その 1 ＞—現状と研究開発の状況—. 情報処理, vol. 54, no. 11, pp. 1164–1170.

Sato, K. (2009). An algorithm for rescheduling freight train locomotives and drivers. *Railway Technology Avalanche*, no. 28, p. 165.

Sato, K. and Fukumura, N. (2009b). Preparation for disruption to freight train traffic. *RRR*, vol. 66, no. 9, pp. 31–34 (in Japanese).
佐藤圭介, 福村直登 (2009b). 貨物列車のダイヤ乱れに備える. RRR, vol. 66, no. 9, pp. 31–34.

Sato, K., Hirai, C. and Fukumura, N. (2009a). A prototype of a crew rescheduling assist system while railway traffic is disrupted. *JREA*, vol. 52, no. 8, pp. 7–10 (in Japanese).
佐藤圭介, 平井力, 福村直登 (2009a). 列車ダイヤ乱れ時の乗務員運用整理業務支援システムのプロトタイプ開発. JREA, vol. 52, no. 8, pp. 7–10.

Sato, K., Hirai, C. and Fukumura, N. (2008). A prototype of a crew rescheduling assist system while railway traffic is disrupted. In *Collection of Papers of the 9th Freight Railway Promotion Award*, pp. 3–21 (in Japanese).
佐藤圭介, 平井力, 福村直登 (2008). 列車ダイヤ乱れ時の乗務員運用整理業務支援システムのプロトタイプ開発. 第 9 回鉄道貨物振興奨励賞受賞論文集, pp. 3–21.

Non-peer-reviewed Conference Papers

Sato, K. (2015b). Mixed integer programming models & algorithms for timetable, rolling stock & crew rescheduling. In *Proceedings of Research Workshop on Railway Operations Modelling, Analysis and Simulation at RTRI*, Kokubunji, Japan, pp. 79–104.

Tamura, K., Tomii, N. and Sato, K. (2013b). Train rescheduling algorithm for minimizing passengers' dissatisfaction by mixed integer programming. In *2013 National Convention Record I.E.E. Japan*, Nagoya, Japan, vol. 3, pp. 128–129 (in Japanese).
田村啓, 富井規雄, 佐藤圭介 (2013b). 旅客損失を最小にする混合整数計画法による運転整理アルゴリズムとその評価. 平成 25 年電気学会全国大会講演論文集, 名古屋, 日本, vol. 3, pp. 128–129.

Chigusa, K., Koseki, T. and Sato, K. (2010a). A passenger-oriented optimization of train rescheduling based on mathematical programming. In *Proceedings of The 17th Jointed Railway Technology Symposium 2010 (J-RAIL2010)*, Tokyo, Japan, pp. 523–526 (in Japanese).
千種健二, 古関隆章, 佐藤圭介 (2010a). 数理計画法に基づく旅客を考慮した異常時列車運行計画の最適化. 第 17 回鉄道技術連合シンポジウム (J-RAIL2010) 講演論文集, 東京, 日本, pp. 523–526.

Chigusa, K., Koseki, T. and Sato, K. (2010b). A passenger-oriented mathematical programming approach for an optimization of train rescheduling. In *The Papers of Joint Technical Meeting on "Transportation and Electric Railway" and "Intelligent Transport Systems," IEE Japan*, Kumamoto, Japan, no. TER-10-57(ITS-10-48), pp. 33–38 (in Japanese).
千種健二, 古関隆章, 佐藤圭介 (2010b). 数理計画法に基づく旅客の観点から見た異常時列車運行計画の最適化. 電気学会研究会資料：交通・電気鉄道ＩＴＳ合同研究会, 熊本, 日本, no. TER-10-57(ITS-10-48), pp. 33–38.

Sato, K., Fukumura, N., Shigeta, E., Kawakami, H., Morita, J. and Watanabe, T. (2009b). A study on a freight train rescheduling automation system (part 2)—Locomotive rescheduling and communication. In *The Collection of Technical Papers on the 46th Railway Cybernetics and Symposium*, Osaka, Japan, Paper ID: 404 (in Japanese).
佐藤圭介, 福村直登, 重田英貴, 川上浩司, 森田潤, 渡辺哲夫 (2009b). 貨物列車運用整理案自動作成システムの研究 (第 2 報)—機関車運用整理機能と整理案伝達機能の開発—. 第 46 回鉄道サイバネ・シンポジウム論文集, 大阪, 日本, 論文番号 404.

Sato, K. and Fukumura, N. (2008). The rolling stock rescheduling problem while traffic is disrupted. In *Abstracts of The 2008 Fall National Conference of Operations Research Society of Japan*, Sapporo, Japan, pp. 192–193 (in Japanese).
佐藤圭介, 福村直登 (2008). 輸送障害時の車両スケジュール変更問題. 日本オペレーションズ・リサーチ学会 2008 年秋季研究発表会アブストラクト集, 札幌, 日本, pp. 192–193.

Shigeta, E., Kawakami, H., Sato, K., Fukumura, N., Morita, J. and Watanabe, T. (2007). A study on a freight train rescheduling automation system—Crew rescheduling. In *The Collection of Technical Papers on the 44th Railway Cybernetics and Symposium*, Tokyo, Japan, Paper ID: 414 (in Japanese).
重田英貴, 川上浩司, 佐藤圭介, 福村直登, 森田潤, 渡辺哲夫 (2007). 貨物列車運用整理案自動作成システムの研究—乗務員運用整理機能の開発—. 第 44 回鉄道サイバネ・シンポジウム論文集, 東京, 日本, 論文番号 414.

Sato, K. and Fukumura, N. (2007a). The crew rescheduling problem while railway traffic is disrupted. In *Handout of SSOR Session, The 2007 Fall National Conference of Operations Research Society of Japan*, Tokyo, Japan (in Japanese).
佐藤圭介, 福村直登 (2007a). 列車ダイヤ乱れ時の乗務員再スケジューリング問題. 日本オペレーションズ・リサーチ学会 2007 年秋季研究発表会 SSOR セッション資料, 東京, 日本.

Sato, K. and Fukumura, N. (2007b). The crew rescheduling problem while railway traffic is disrupted. In *Proceedings of Summer Seminar on Operations Research (SSOR) 2007*, Ito, Japan, no. 10-3 (in Japanese).
佐藤圭介, 福村直登 (2007b). 列車ダイヤ乱れ時の乗務員再スケジューリング問題. 日本オペレーションズ・リサーチ学会 SSOR 2007 アブストラクト集, 伊東, 日本, no. 10-3.