May 2014

Operations Research Modeling of Cyclic Train Timetabling, Cyclic Train Platforming, and Bus Routing Problems

Mojtaba Heydar
University of Wisconsin-Milwaukee

OPERATIONS RESEARCH MODELING OF CYCLIC TRAIN TIMETABLING, CYCLIC TRAIN PLATFORMING, AND BUS ROUTING PROBLEMS

by

Mojtaba Heydar

A dissertation Submitted in

Partial Fulfillment of the

Requirement for the Degree of

Doctor of Philosophy

in Engineering

at

The University of Wisconsin – Milwaukee

May 2014

ABSTRACT

OPERATIONS RESEARCH MODELING OF CYCLIC TRAIN TIMETABLING, CYCLIC TRAIN PLATFORMING, AND BUS ROUTING PROBLEMS

by

Mojtaba Heydar

The University of Wisconsin-Milwaukee, 2014
Under the supervision of Professor Matthew Petering

Public transportation or mass transit involves the movement of large numbers of people between a given numbers of locations. The services provided by this system can be classified into three groups: (i) short haul: a low-speed service within small areas with high population; (ii) city transit: transporting people within a city; and (iii) long haul: a service with long trips, few stops, and high speed (Khisty and Lall, 2003). It can be also classified based on local and express services. The public transportation planning includes five consecutive steps: (i) the network design and route design; (ii) the setting frequencies or line plan; (iii) the timetabling; (iv) the vehicle scheduling; and (v) the crew scheduling and rostering (Guihaire and Hao, 2008; Schöbel, 2012).

The first part of this dissertation considers three problems in passenger railway transportation. It has been observed that the demand for rail travel has grown rapidly over the last decades and it is expected that the growth continues in the future. High quality railway services are needed to accommodate increasing numbers of passengers and goods. This is one of the key factors for economic growth. The high costs of railway infrastructure ask for an increased utilization of the existing infrastructure. Attractive

railway services can only be offered with more reliable rolling stock and a more reliable infrastructure. However, to keep a high quality standard of operations, smarter methods of timetable construction are indispensable, since existing methods have major shortcomings.

The first part of this dissertation, comprising Chapters 1-6, aims at developing a cyclic (or periodic) timetable for a passenger railway system. Three different scenarios are considered and three mixed integer linear programs, combined with heuristics for calculating upper and lower bounds on the optimal value for each scenario, will be developed. The reason of considering a periodic timetable is that it is easy to remember for passengers. The main inputs are the line plan and travel time between and minimum dwell time at each station. The output of each model is an optimal periodic timetable.

We try to optimize the quality of service for the railway system by minimizing the length of cycle by which trains are dispatched from their origin. Hence, we consider the cycle length as the primary objective function. Since minimizing travel time is a key factor in measuring service quality, another criterion—total dwell time of the trains—is considered and added to the objective function.

The first problem, presented in Chapter 3, has already been published in a scholarly journal (Heydar et al., 2013). This chapter is an extension of the work of Bergmann (1975) and is the simplest part of this research. In this problem, we consider a single-track unidirectional railway line between two major stations with a number of stations in between. Two train types—express and local—are dispatched from the first station in an alternate fashion. The express train stops at no intermediate station, while the local train should make a stop at every intermediate station for a minimum amount of dwell time. A mixed integer linear program is developed in order to minimize the length of the dispatching cycle and minimize the total dwell time of the local train at all stations combined. Constraints include a minimum dwell time for the local train at each station, a maximum total dwell time for the local train, and headway considerations on the main line an in stations. Hundreds of randomly generated problem instances with up to 70 stations are considered and solved to optimality in a reasonable amount of time. Instances of this problem typically have multiple optimal solutions, so we develop a procedure for finding all optimal solutions of this problem.

In the second problem, presented in Chapter 4, we present the literature's first mixed integer linear programming model of a cyclic, combined train timetabling and platforming problem which is an extension of the model presented in Chapter 3 and Heydar et al. (2013). The work on this problem has been submitted to a leading transportation journal (Petering et al., 2012). From another perspective, this work can be seen as investigating the capacity of a single track, unidirectional rail line that adheres to a cyclic timetable. In this problem, a set of intermediate stations lies between an origin and destination with one or more parallel sidings at each station. A total of *T* train types—each with a given starting and finishing point and a set of known intermediate station stops—are dispatched from their respective starting points in cyclic fashion, with one train of each type dispatched per cycle. A mixed integer linear program is developed in order to schedule the train arrivals and departures at the stations and assign trains to tracks (platforms) in the stations so as to minimize the length of the dispatching cycle and/or minimize the total stopping (dwell) time of all train types at all stations combined. Constraints include a minimum dwell time for each train type in each of the stations in which it stops, a maximum total dwell time for each train type, and headway considerations on the main line and on the tracks in the stations. This problem belongs to the class of NP-hard problems. Hundreds of randomly generated and real-world problem instances with 4-35 intermediate stations and 2-11 train types are considered and solved to optimality in a reasonable amount of time using IBM ILOG CPLEX.

Chapter 5 expands upon the work in Chapter 4. Here, we present a mixed integer linear program for cyclic train timetabling and routing on a single track, bi-directional rail line. There are T train types and one train of each type is dispatched per cycle. The decisions include the sequencing of the train types on the main line and the assignment of train types to station platforms. Two conflicting objectives—(1) minimizing cycle length (primary objective) and (2) minimizing total train journey time (secondary objective)—are combined into a single weighted sum objective to generate Pareto optimal solutions. Constraints include a minimum stopping time for each train type in each station, a maximum allowed journey time for each train type, and a minimum headway on the main line and on platforms in stations. The MILP considers five aspects of the railway system: (1) bi-directional train travel between stations, (2) trains moving at different speeds on the main line, (3) trains having the option to stop at stations even if they are not required to, (4) more than one siding or platform at a station, and (5) any number of train types. In order to solve large scale instances, various heuristics and exact methods are employed for computing secondary parameters and for finding lower and upper bounds on the primary objective. These heuristics and exact methods are combined with the math model to allow CPLEX 12.4 to find optimal solutions to large problem instances in a reasonable amount of time. The results show that it is sometimes necessary for (1) a train type to stop at a station where stopping is not required or (2) a train type to travel slower than its normal speed in order to minimize timetable cycle time.

In the second part of this dissertation, comprising Chapters 7-9, we study a transit-based evacuation problem which is an extension of bus routing problem. This work has been already submitted to a leading transportation journal (Heydar et al., 2014). This paper presents a mathematical model to plan emergencies in a highly populated urban zone where a certain numbers of pedestrians depend on transit for evacuation. The proposed model features a two-level operational framework. The first level operation guides evacuees through urban streets and crosswalks (referred to as “the pedestrian network”) to designated pick-up points (e.g., bus stops), and the second level operation properly dispatches and routes a fleet of buses at different depots to those pick-up points and transports evacuees to their destinations or safe places. In this level, the buses are routed through the so-called “vehicular network.” An integrated mixed integer linear program that can effectively take into account the interactions between the aforementioned two networks is formulated to find the maximal evacuation efficiency in the two networks. Since the large instances of the proposed model are mathematically difficult to solve to optimality, a two-stage heuristic is developed to solve larger instances of the model. Over one hundred numerical examples and runs solved by the heuristic illustrate the effectiveness of the proposed solution method in handling large-scale real-world instances.

To
My parents
Fatima and Mohammad

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENT

I would like to thank a number of people who have supported me in any way during my Ph.D. study. First of all, I would like to thank my Ph.D. advisor and committee chair, Dr. Matthew Petering for being my advisor, for patiently helping me to construct and finalize my Ph.D. research, for assisting me to provide better outputs, for challenging me with different ideas and methods, and for his tireless effort and continuous support. Throughout this year, he has been very inspirational to me and has been a great friend and great mentor. I would also like to thank to my dissertation committee memebrs: Dr. Jaejin Jang, Dr. Wilkistar Otieno, Dr. Alan Horowitz, and Dr. Gabriella Pinter for their evaluation, invaluable directions, and insightful comments. I have learned a great deal from them and I feel very much indebted. The second part of this research is the work I started with Dr. Troy Liu from Civil Engineering and Mechanics Department. I would like to thank him for giving me the opportunity to work on this problem. I am grateful for the College of Engineering and Applied Science at University of Wisconsin-Milwaukee for providing me the financial support. I would like to thank to the faculty in Industrial and Manufacturing Engineering Department for their invaluable assistance in my training. I never will forget the good memories that I had here. I would like to acknowledge Dietrich Bergamnn who passed away in 2012 while we were preparing our first manuscript for submission. Dietrich's innovative ideas inspired us to further develop the mathematical models presented in this dissertation. Finally, I am greatly appreciative of my parents and other members of my family for their infinite love and support throughout my life and while I have been abroad for my PhD studies.

Part I

Cyclic Train Timetabling and Platforming

Chapter 1

Introduction

1.1. Motivation

Transportation plays an important role in modern societies. Rail transport consists of moving goods or passengers using railroads or railways. A railroad is composed of two parallel rails attached perpendicularly to beams called "crossties" or simply "ties" to keep a constant distant apart. The vehicles moving over the rails are arranged in a train: a set of vehicles coupled together. These vehicles are typically referred to as rolling stock. These vehicles may be powered or un-powered. Powered vehicles are referred to as locomotives while un-powered vehicles are referred to as cars, carriages, wagons or coaches (for passengers).

Railroads provide the most energy efficient and cost-effective transportation services over land since, compared to vehicles on paved roads, railcars make much less friction when moving over rails. As a result, trains typically use less energy than road vehicles to transport a given tonnage of freight or a given number of passengers over a given distance. Nevertheless, rail is still a capital-intensive means of transport (Luthi, 2009).

The problems that railroad companies are facing are generally more complex than those found in other modes of transportation such as trucking and air transportation. This is due mainly to the enormous, interrelated decisions involved (Ahuja et al., 2005). Many rail transportation problems are interesting combinatorial optimization problems. This difficulty has attracted operations research (OR) practitioners over the past two decades which has resulted in numerous articles published in operations research journals. However, the U.S. railroad industry has not implemented many of these advances (Ahuja et al., 2005).

Railway systems can be divided into two types: passenger and freight. Passenger railway systems are the focus of this dissertation and are becoming increasingly competitive with air transportation. High speed rail (HSR) is one type of passenger rail that has been receiving enormous attention over the last three decades. HSR is defined differently in different countries. In the European Union, a line is defined high-speed if it is built for speeds greater than or equal to 250 km/hr (150 mph) or is upgraded with speeds greater than 200 km/hr (124 mph) (International Union of Railways, 2009; Feigenbaum, 2013). The term is defined in U.S. quite differently. In this definition HSR has three classes: emerging, with a speed of 90-110 mph; regional, with a speed of 110-150 mph; and express with speeds exceeding 150 mph (Feigenbaum, 2013). Campos and de Rus (2009) classify four types of high-speed railway services based on their relationship with conventional services:

1) The dedicated or exclusive exploitation models in which high-speed rail and conventional rail have separate infrastructure. Japan's Shinkansen uses this model. It was developed because the conventional infrastructure had reached its full capacity and the track was not able to support high-speed traffic due to its narrow gauge. The advantage of this model is that market organizations of these two services are independent.
2) The mixed high-speed models that include both dedicated high-speed track and upgraded conventional track. Dedicated tracks serve high-speed trains

only, while upgraded tracks serve both high-speed and conventional trains. France's TGV (Train à Grande Vitesse) is an example of this model. The reduced building cost is the main advantage of this model.

3) The mixed conventional models consist of upgraded tracks—used by both high-speed and conventional traffic—and conventional tracks that serve only conventional trains. Spain's AVE (Alta Velocidad Española) is an example of this mode.
4) The fully mixed models in which both high-speed and conventional trains can use each type of infrastructure. The Inter City Express (ICE) in Germany is an example of this model. This model allows full flexibility depending on track availability the rolling stock can use it with its corresponding speed.

High speed rail (HSR) is becoming more desirable as it aims at reducing highway and airport congestion, cutting national dependence on foreign oil, and improving rural and urban environments by reducing carbon emissions. For these reasons countries such as Japan, France, Germany, Spain, and Italy have developed large HSR networks in recent years (Albalate and Bel, 2012). China is the country with the most recent HSR network. A recent study by Albalate and Bel (2012) mentions that high speed trains now travel the 644-mile distance between Wuhan and Guangzhou in about three hours, much less than the ten hours that were previously needed to travel between these two major cities.

Table 1.1 lists all countries that have high-speed railway systems in operation as of late 2013 (UIC High Speed Department, 2013). Japan's Shinkansen was the first high-speed railway system in the world and was built in 1964 to connect the cities of Tokyo and Osaka (Feigenbaum, 2013). The system now includes seven lines operated by four operators: JR Central; JR West; JR East; and JR Kyushu. The current network features 2664 kilometers of track with maximum speeds ranging from 245 to 320 km/hr. The system has 779 km of network in construction with 179 km planned for the future (UIC High Speed Department, 2013).

Italy opened the second world's high-speed railway line between Rome and Florence in 1977 (Feigenbaum, 2013). The Italian network now features 923 kilometers of track with maximum speeds between 250 and 300 km/hr. The services over the high-speed lines are provided by Nuovo Transporto Viaggiatori or NTV—a private owned company—and TrenItalia.

After Japan and Italy, France developed the world's third high-speed railway system. It is referred to as TGV (Train à Grande Vitesse) and operated by SNCF. The network now has 2036 kilometers of track in operation and connects many cities across France and adjacent countries with a maximum speed between 300 and 320 km/hr making it the fastest high-speed train in Europe.

Germany's high-speed rail system opened in 1991 which was encouraged mainly by France's TGV and the high-speed rail line in Italy. Unlike the French network, the German ICE connects many different hubs resulting in more stops. The network includes eleven different train lines and 1334 kilometers of upgraded and newly built track with maximum speeds ranging from 230 to 300 km/hr (Feigenbaum, 2013; UIC High Speed Department, 2013).

Following Germany's ICE, Spain opened its high-speed rail line—AVE—in 1992 (Feigenbaum, 2013). The network now features 2515 kilometers of track with a maximum speed of 200 – 300 km/hr (UIC High Speed Department, 2013). Other countries in Europe that have more recently developed high-speed railway systems include Austria, Belgium, The Netherlands, Switzerland, and the UK (UIC High Speed Department, 2013).

In recent years the following countries in Asia have also developed high-speed railway systems: China, South Korea, Taiwan, Turkey, and Uzbekistan. According to Feigenbaum (2013), China's high-speed rail network is now the largest in the world. The network currently includes 9867 kilometers of track with another 9081 kilometers under construction. South Korea opened its high-speed railway system in 2004 with a single line connecting Seoul to Daegu. The second line which connects Daegu to Pusan started operating in 2010. Taiwan's high-speed railway system, which began operating in 2007, includes a 345-kilometer line connecting Taipei to Kaohsiung with a maximum speed of 300 km/h. In 2009, Turkey opened a 232-kilometer long high-speed rail line between Ankara and Eskisehir with 250 km/hr as the maximum speed.

These successful high-speed rail implementations around the globe, together with various economic, mobility, and environmental concerns, are now being used to justify the construction and implementation of new HSR networks in other nations. Based on these considerations, in April 2009 the U.S. government unveiled its blueprint for a national HSR network. Table 1.2 displays the expected demand for the first four HSR lines in the United States.

Table 1.1 List of high speed railway systems around the world (source: UIC High Speed Department, 2013)

Country	In operation (km)	Under construction (km)	Total country (km)
Austria	93	0	93
Belgium	209	0	209
China	9867	9081	18948
France	2036	757	2793
Germany	1334	428	1762
Italy	923	0	923
Japan	2664	779	3443
Netherlands	120	0	120
South Korea	412	186	598
Spain	2515	1308	3823
Switzerland	35	72	107
Taiwan	345	0	345
Turkey	444	603	1047
United Kingdom	113	0	113
United States	362	0	362
Uzbekistan	344	0	344

Subways and light rail systems are other types of passenger railway systems that are used to transport large numbers of passengers within a city. These systems are sometimes referred to as rapid transit systems, undergrounds, or metros. Table 1.3 lists all metro systems around the world that are in operation and have more than 100 stations.

Table 1.2 Expected demand for the first four U.S. high speed railway route (source: Albalate and Bel, 2012)

Route.	Maximum Speed	Miles	Annual Ridership (one-way trips)	Year forecast
San Francisco-Los Angeles	220 mph	500	7.2 million	2035
Chicago-St. Louis	220 mph	297	2.1 million	2035
Orlando-Tampa	186 mph	85	1.6 million	2035
Albany-New York City	220 mph	142	2.3 million	2035

Freight railway systems are the other type of railway system. Although not the focus of this dissertation, they are vital to the U.S. economy. According to the American Association of Railroads, the freight railway industry moves over 30 million carloads per year in almost 500,000 railroad-owned railcars earning more than $50 billion per year in revenue (AAR, 2007). This activity accounts for approximately one-third of the total freight moved in the US each year when measured in ton-miles (Gorman et al., 2011). The North American freight rail industry is composed of seven major railroads and several hundred regional and short-line carriers that span 120,000 track miles. For instance, as pointed out by Ahuja et al. (2005), a typical Class I U.S. railroad owns more than 10,000 miles of track, 2,000-3,000 locomotives, about 80,000 railcars, 200-300 classification yards, and employs more than 5,000 crew members to operate thousands of trains.

Table 1.3 List of subway in the world with more than 100 stations (source: Wikipedia)

Location	Country	Name	Year opened	Stations	System length	Year of last extension
Vienna	Austria	Vienna U-Bahn	1976	104	80 km (50 mi)	2013
Santiago	Chile	Santiago Metro	1975	108	103 km (64 mi)	2011
Beijing	China	Beijing Subway	1981	232	465 km (289 mi)	2014
Guangzhou	China	Guangzhou	1997	130	240 km (150 mi)	2013
Hong Kong	China	Mass Transit Railway	1979	152	218 km (136 mi)	2013
Shanghai	China	Shanghai Metro	1993	263	538 km (334 mi)	2013
Shenzhen	China	Shenzhen Metro	2004	131	178.4 km (110.9 mi)	2011
Paris	France	Paris Metro	1900	303	214 km (133 mi)	2013
Berlin	Germany	Berlin U-Bahn	1902	170	151.7 km (94.3 mi)	2009
		Berlin S-Bahn	1924	166	332 km (206 mi)	2007
Delhi	India	Delhi Metro	2002	141	190 km (118.1 mi)	2011
Milan	Italy	Milan Metro	1964	103	94.5 km (58.7 mi)	2014
Osaka	Japan	Osaka Municipal Subway	1933	101	137.8 km (85.6 mi)	2006
Tokyo	Japan	Toei Subway	1960	106	121.5 km (75.5 mi)	2000
		Tokyo Metro	1927	179	195.1 km (121.2 mi)	2008
Busan	S Korea	Busan Metro	1985	128	130.2 km (80.9 mi)	2011
Seoul Metropolitan Area	S Korea	Seoul Subway (Lines 1-9)	1974	296	327 km (203.2 mi)	2012
Mexico City	Mexico	Mexico City Metro	1969	195	226.5 km (140.7 mi)	2012
Oslo	Norway	Oslo Metro	1966	105	84.2 km (52.3 mi)	2006
Moscow	Russia	Moscow Metro	1935	194	325.4 km (202.2 mi)	2014
Singapore	Singapore	Mass Rapid Transit	1987	106	148.7 km (92.4 mi)	2013
Barcelona	Spain	Barcelona Metro	1924	165	123.7 km (76.9 mi)	2011
Madrid	Spain	Madrid Metro	1919	300	293 km (182 mi)	2010
Stockholm	Sweden	Stockholm Metro	1950	100	105.7 km (65.7 mi)	1994
Taipei	Taiwan	Taipei Metro	1996	103	121.3 km (75.4 mi)	2013
London	United Kingdom	London Underground	1890	270	402 km (250 mi)	2008
Chicago	United States	Chicago ‘L’	1897	145	165.4 km (102.8 mi)	1993
New York City	United States	New York City Subway	1904	421	373 km (232 mi)	2013

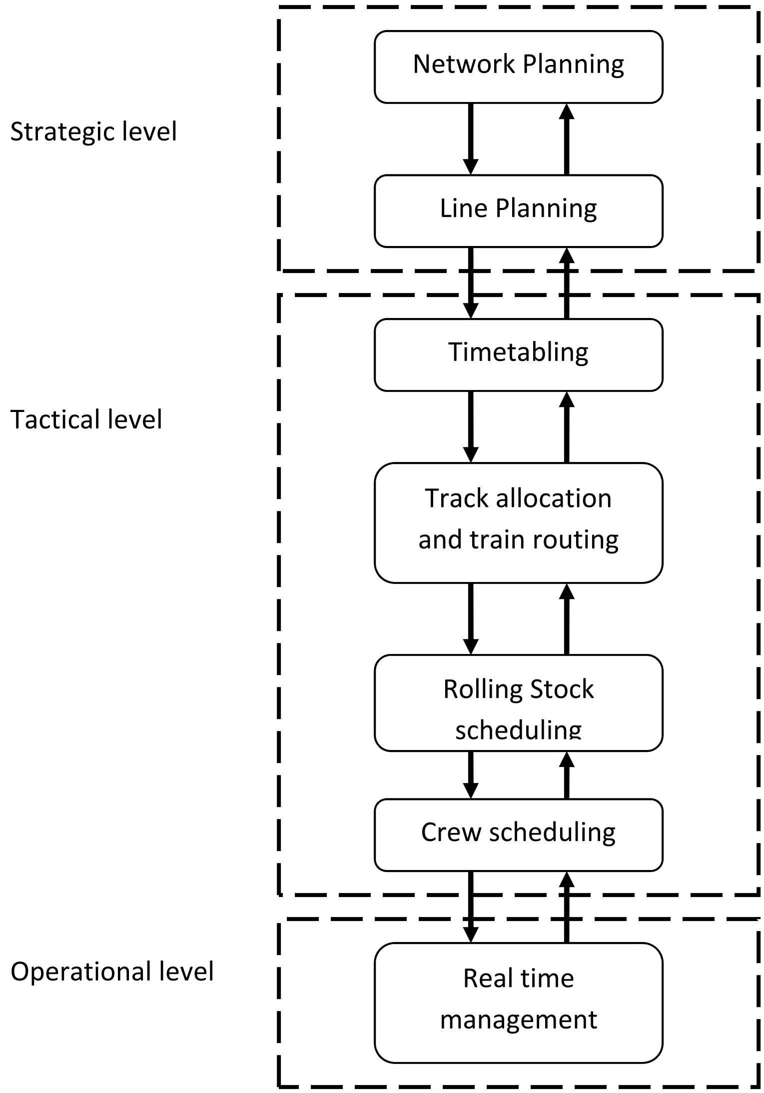


Figure 1.1 Hierarchy of decision problems in the railway planning process (source: Lusby et al., 2011a).

1.2. The railway planning process

A railway system consists of the following five components:

1. The infrastructure on which the trains run including its power supply, telecommunication systems, safety systems, and traffic control systems.

2. The rolling stock which consists of locomotives, electrical multiple units (EMUs), and diesel multiple units (DMUs) (Goverde, 2005).
3. The schedule which determines the arrival (departure) time of trains at (from) stations and/or junctions.
4. The railroad employees
5. The operational rules

Planning railway transport and optimizing the use of railway infrastructure are highly complex tasks. Passenger railway companies divide the planning process into a hierarchical process with several phases (Bussieck, 1998). One way as noted by Lusby et al. (2011a) is to divide the problems, and associated decisions, into strategic, tactical, and operational phases (Figure 1.1). The time horizon for strategic decisions is 5-15 years and these decisions are often regarded as resource acquisition problems. Tactical decisions typically consider a 0.5 to 5 year planning horizon and their main goal is to allocate resources effectively and efficiently. Optimum resource allocation can make a difference between profit and loss for a railway transport company. Operational decisions are short term decisions and may have a planning horizon from 1 hour to several months. The main goal of these decisions is to manage resource consumption optimally (Bussieck et al., 1997; Schlechte, 2012).

The main problems at the strategic level are network planning—when and how to invest in railway infrastructure—and line planning—deciding train origins, destinations, and frequencies. The railway network design problem takes railway transport and traffic demand into account. This problem and its associated decisions are costly and will have economic and environmental impacts which may last for several decades. Train types that operate on a network can be homogeneous or heterogeneous (Goverde, 2005). A line is a route in a railway network that connects two terminal stations, called the origin and destination. The line planning problem (LPP) corresponds to the problem of selecting the set of lines and their frequencies with optimizing two objectives: minimizing the operational cost and maximizing the number of travelers with direct connections (Bussieck et al., 1996; Claessens et al., 1998; Goossens et al., 2004; Caprara et al., 2007).

Timetabling is one of the four main tactical problems that are solved during the overall railway planning process (Figure 1.1). Other problems at this level include track allocation and train routing, rolling stock scheduling, and crew scheduling. A timetable specifies points in time and assigns them to a train's arrival (departure) time at (from) several consecutive stations. Therefore, it connects the train line plan to the available infrastructure. The fundamental domains, structures, and principles of railway timetable planning and signaling are introduced in Section 1.3.

Other decisions at the tactical level include track allocation and train routing, rolling stock scheduling, and crew scheduling. The purpose of track allocation and train routing is to implement the timetable that has just been developed considering infrastructure and the capacity on the main line, at stations, at sidings, and at junctions. The rolling stock decision assigns physical train units to the trains listed in the timetable considering the feasibility of this assignment during peak hour operations. Crew scheduling is the last decision at tactical level and assigns crew to the train units. This is an NP-hard problem (Goverde, 2005). Nowadays, the infrastructure, rolling stock, and operational rules are designed so that passengers and goods can be transported to their destination as safely, quickly, punctually, cheaply, and comfortably as possible.

1.3. Railway timetables

After designing the line plan, the next step in planning process is to establish a timetable. A timetable (i.e. train schedule) specifies points in time and assigns them to train arrival (departure) events at stations or sidings. Train timetables and schedules have a significant impact on infrastructure utilization and punctuality. Decisions taken during operations are often based on the timetable. In principal two general methods are theoretically possible for railway operations: based on a detailed timetable or without a timetable. Regarding passenger trains, a timetable for customer information and trip planning is needed. On the other hand, demand for freight transport can change rapidly and requests for a train path are often short term. Consequently, rail freight traffic can sometimes be operated without a schedule even if most other trains are operated on the basis of an exact timetable. The schedule has a major impact on the competitiveness of transport and determines expenditure. Therefore, short and reliable travel times are needed to attract customers.

A timetable is developed in several phases starting possibly with a conceptual idea about 10-20 years before implementation and ending as input for daily operation. The timetable in use is strongly connected with defining schedules for staff and rolling stock. In particular, during daily operations, a new schedule must ensure that, after incident or delay, the rolling stock and staff are ready to run a consecutive or new service (Luthi, 2009). The final schedule accounts for the following time components:

- The technical minimum run and dwell times
- Running time and dwell time supplements

- Time supplements (or constraints) for connections as part of the integrated fixed-interval timetable
- Buffer times between trains, also referred to as headway.

The timetable is a crucial factor for the success of passenger railways. Frequent, fast, and direct connections are desirable but not always possible. The service is thus optimized through coordinated connections.

Generating a new timetable is a complex process. Even with the aid of computer programs, calculating a new timetable from scratch for an entire, complex network within a short time is not possible today. As a result, new timetables are usually modified based on the previous year's timetable in critical sections by shifting train paths by a few minutes (Luthi, 2009). Chapters 2-6 of this dissertation will explore passenger train timetabling issues in detail.

1.4. Related railway issues

In this section we briefly discuss three issues related to railway timetabling systems: train control systems, passenger demand, and disruption management.

1.4.1. Train control systems

Safety is a very important factor in any railway system. Therefore, railway systems are equipped with safety systems through train control technology and track signaling systems. Track signaling systems are partitioned into two categories: automatic signals and controlled signals. Trains moving on a track may be separated by automatic block signaling and/or automatic train protection systems (Goverde, 2005). In railroad signal terminology, a *block* is a segment of track with predefined limits and typically can range from half a mile to two miles long. There are two types of block signaling systems: fixed block signaling system and moving block signaling system. In the fixed block signaling system, each stretch of open track is divided into fixed block sections that are controlled by a signal governing train entrance into the block. The block signal at the entrance of an open track determines whether or not the block can be used by an entering train. In order to use the block signal, the train location should be determined by the system. In modern railway systems the presence of a train in a block is detected using train detection devices such as *track circuits*. A track circuit is connected to the signal via electric wire at its both ends. Automatic block signaling (ABS) systems protect trains against head-tail collisions. In the moving block systems, train locations are determined in real time based on the train performance characteristics. Most HSR systems use a different signaling system. In these systems, the communication equipment is fully integrated in the cab which makes signal control along the rail line unnecessary (Campos and de Rus, 2009). The most recent technology of train control systems is called positive train control (PTC) system. The U.S. Congress is requiring the implementation of PTC by all railway company across most of the U.S. by December 31, 2015 (Peters and Frittelli, 2012). As defined by Association of American Railroad (accessed May 7, 2014), "*Positive train control (PTC) is advanced technology designed to automatically stop or slow a train before certain accidents occur. In particular, PTC is designed to prevent train-to-train collisions, derailments caused by excessive speed, and unauthorized movement of trains onto sections of track where repairs are being made or as a result of a misaligned track switch*." The PTC system works according the following steps (Chertock and Diaz, 2012):

- An on-board computer is initialized before a train departs its origin.
- The train's location is determined by GPS in conjunction with a geographic track database.
- The on-board computer calculates the warning and braking curve as the train moves on the line.
- As the train is approaching down the track, the on-board computer connects to line-side devices for checking broken rails and signal aspects.

This dissertation assumes the presence of the latter type of signaling system and assumes there is no block signaling system.

1.4.2. Passenger demand

Passenger demand is another issue with importance for passenger train timetabling. The demand for HSR is increasing due to the quality of service and flexibility it provides for passengers. Other factors of increasing demand include improved accessibility (Chang and Lee, 2008; Masson and Petiot, 2009) and potential development for the regions along the corridor between major cities (Lutter et al., 1997; Román et al., 2007). According to de Rus and Nombela (2007), HSR demand is very sensitive to the length of the line and population of cities among the line which means more people use the HSR service.

The people that choose HSR are coming from other traffic modes (e.g. air and auto), other train services (e.g. conventional), or are choosing it as their first choice due to increased wealth due to economic growth (Cascetta and Coppola, 2012). Park and Ha (2006) considers the diversion from air to HSR and analyze the impact of Korean high-speed rail—KTX—on air transport demand between Seoul and Daegu, a short-haul distance, using three variables: access and egress time, fare levels, and operational frequency. Their results show that only 14% of passengers would prefer air travel which is mainly due to the short travel time, faster check-in processes, and more frequent service offered by KTX. This competitiveness is the result of HSR's ability to offer short travel times between cities, reduce access times to the economic centers, handle large passenger volume, and better adjust to peaks or shocks in daily demand (Román et al., 2007).

Two main factors impact demand for their transportation: (1) income and population growth, and (2) the travel environment improvement itself (Yao and Morikawa, 2005). In their study, Yao and Morikawa (2005) investigate an integrated intercity travel demand model when service level of the transportation mode changes substantially. As a case study, an intercity high speed rail project in Japan is considered assuming that trip generation, destination choice, mode choice, and route choice are in the model. The results of the study show that induced travel demand increases when travel time, travel cost, and access time decreases. Also, increasing service frequency will increase travel demand as well. The results also show that business travel demand is more sensitive to the factors just mentioned than non-business travel demand.

As mentioned, accessibility is one of the factors that impact HSR demand. Gutiérrez (2001) analyzes the accessibility impact of the high-speed line connecting Madrid, Barcelona, and the French border by means of three indicators: weighted average travel time, economic potential, and daily accessibility. Among these three indicators, daily accessibility is the most important since it measures relationships over the short distance nodes.

1.4.3. Disruption management

Railway systems are subject to disruptions because they are very complex systems. In this context, the word disruption can refer to any unexpected event. According to Goverde (2005), there are two types of delays that cause disruptions. Primary delays are caused by disruptions within the process, e.g. failing switches, whereas secondary delays are caused by other trains in the form of a train conflict which necessitates waiting for another train. There is a distinction between these two delays: the former is related to infrastructure; while the latter is mainly related to process and planning. However, these two delays are somehow interconnected. That is, improving infrastructure such as the signaling system or adding overtaking points and/or doubling the track segments will create more room for train passing, thereby reducing or eliminating secondary delays. There are six types of secondary delay. These delays occur when (i) a leading train is a slow train, (ii) two trains are assigned to the same platform or share the same route through stations, (iii) a block or track is occupied by another train so the following train must stop until the block or track become clear, (iv) passengers transfer between two trains, (v) rolling stocks are connected, and (vi) crews transfer between two trains or trips. Researchers have proposed several approaches for dealing with disruptions including contingency planning, stochastic models, robust optimization, recoverable robustness, and pure reparation rescheduling (Acuna-Agost, 2010).

1.5. Contribution and novelty of the Research

The first part of this dissertation focuses on operations research modeling of cyclic timetabling, cyclic platforming, and capacity optimization problems for passenger railways. Among these topics, we focus most on cyclic timetable optimization.

This research contributes to the existing literature on train timetabling, train platforming, and the application of operations research techniques in the railway industry in several ways. This research is the first work that (i) models a cyclic train timetabling problem using a mixed integer linear program (MILP) in which the cycle length, an important value that is considered as a parameter in all published works, is the primary objective function to be minimized and is a decision variable; (ii) investigates and maximizes railway capacity from a non-traditional perspective; (iii) presents optimized cyclic train timetables for real-world high-speed railway systems including the Japanese Shinkansen and the Taiwanese high speed railway system using state-of-the-art modeling techniques, heuristic methods, and optimization software; (iv) presents an MILP model that integrates cyclic train timetabling and cyclic train platforming decisions; and (v) generates cyclic timetable and platforming schedules for a bi-directional, single track rail line with heterogeneous rolling stock.

1.6. Organization of Part I

The remainder of Part I of this dissertation is organized as follows. Chapter 2 surveys the applications of operations research methods in railway transportation. In Chapter 3, we study cyclic train timetabling for a single track, unidirectional rail line with two trains types –express and local. A mixed integer linear program is developed to minimize the dispatching cycle of trains and the total dwell time of all train types at all stations combined. This work has already been published in Heydar et al. (2013).

In Chapter 4, a mixed integer linear program (MILP) for cyclic train timetabling and platforming for a single track, unidirectional rail line with homogeneous rolling stock is developed. The objective function is to maximize line capacity through cycle time minimization, and to minimize the total dwell time of all train types at all stations combined. In this problem it is assumed that there are multiple train types that can start or end their journey at stations other than the terminal stations. Also, the model determines the train routing—i.e. the platform assignment—at stations with more than one platform. In other words, this is the literature's first MILP model to integrate cyclic train timetabling and train platforming problems. The model is then validated by solving two real-world examples taken from the Taiwanese and Japanese high speed railway systems.

In Chapter 5, a more complex model for cyclic, combined train timetabling and platforming is considered for a bi-directional, single track railway line between two terminals with heterogeneous rolling stock. The proposed mixed integer linear program is developed to minimize cycle time and minimize total train journey time. The model also determines platform assignments at stations with more than one platform or track. It is assumed that trains can move slower than full speed. Also, another feature of this model is that some trains may stop at stations or sidings if their line plan does not require them to do so. This increases the model flexibility as well as complexity.

Chapter 6 summarizes the work that has been conducted and lays the groundwork for future work in this area.

Chapter 2

Review of literature of railway problems

In this chapter we review the literature on railway operations that considers train timetabling, train routing, railway capacity analysis, or the railway literature itself. Excellent surveys of the railway operations literature have been done by Harrod (2012), Lusby et al. (2011a), Abril et al. (2008a), Caprara et al. (2007), Huisman et al. (2005), Cordeau et al. (1998), and Bussieck et al. (1997). A comprehensive overview of railway timetabling and traffic issues is provided by Hansen and Pachl (2008).

Railway systems can be classified based on their purpose. Articles on railway operations may be divided into two main categories—those emphasizing passenger railway systems and those emphasizing freight railway systems.

2.1. Freight railway systems

Examples of recent research on freight railway systems include Ahuja et al. (2005), Crainic et al. (1990), Gorman et al. (2010), and Verma et al. (2011). The problems that can be addressed, and have been studied so far, include, but are not limited to, railroad blocking problem, train scheduling, classification yard location, train dispatching, locomotive scheduling, and crew scheduling. In this section the classification yard location problem and railroad blocking problem are discussed since they are unique to freight railway systems. Other problems will be discussed in Section 2.2 as they are common to both freight and passenger railway systems.

2.1.1. Classification yard location problem

Classification yards are the intermediate nodes in a national rail network where railcars are attached to and detached from each other to form trains and train units called blocks. Yards are classified into three types: local, system, and regional. As pointed out by Ahuja et al. (2005) a major U.S. railroad has about 20-40 hub yards. The yard location problem considers where to locate new yards and where to close exiting yards (Ahuja et al., 2005). According to Ahuja et al. (2005), typical questions in this regard that can be answered through OR include:

(i) What yards can be shut down with minimal impact on the transportation cost?

(ii) What is the optimal trade-off between maximizing the number of yards and minimizing transportation costs?

(iii) The locations and number of new yards in case of expansion.

(iv) What is the best network configuration?

2.1.2. Railroad Blocking Problem

The railroad blocking problem (RBP) is one of the most important problems in freight railway operations. A solution to this problem specifies how to combine a large number of shipments into a block, i.e. a large portion of railcars that are linked to each other within a larger train, so as to reduce their individual handling as they travel from an origin to a destination. The railroad blocking problem is to construct a network of blocks in order to minimize the total transportation cost when all rail shipments are routed over this blocking network. The objective function is the weighted sum of total travel cost and intermediate handling cost (Ahuja et al. 2005). Researchers have modeled and solved this planning problem using network models (Newton et al. 1998), mixed integer linear programs (Barnhart et al, 2000) and heuristic algorithms such as very large-variable neighborhood search (VLNS) algorithm (Ahuja et al., 2007).

2.2. Passenger railway systems

Researchers have classified the planning decisions in passenger railway system in different ways. One way as noted by Lusby et al. (2011a) is to divide the problems into strategic, tactical, and operational planning phases (Figure 1.1). The main problems at the strategic level are network planning—when and how to invest in railway infrastructure—and line planning—deciding train origins, destinations, and frequencies. At the tactical level, problems related to timetabling, track allocation and train routing, rolling stock scheduling, and crew scheduling are considered and solved.

Operational planning mainly deals with real-time management of railway traffic. Other surveys that classified railway problems include Bussieck et al. (1997), Cordeau et al. (1998), Huisman et al. (2005), and Caprara et al. (2007). In what follows, the problems and literature related to each level of decisions will be presented. Although these decisions have been classified as passenger railway decision, the same decisions can be defined and considered for freight as well. For instance, train scheduling and timetabling problem can be defined for both systems. This classification is important since it enables us to compare the cyclic and noncyclic train timetables.

2.2.1 Strategic decisions

As defined, a strategy is a long-term plan which is made to achieve a certain objective and is concerned with the development of the network and long-term acquisition of resources. These decisions include network planning, rolling stock, crew planning and line planning.

The network planning problem deals with the design of railroad networks. This problem concerns construction of a new infrastructure and/or expansion or modification of an existing infrastructure such as addition of a second track (Lusby et al., 2011a). This problem can be solved through facility location techniques; hence they are very complex problems. Rolling stock acquisition corresponds to the decision of buying, selling, leasing, hiring, or wasting locomotives and cars. This problem is considered as a strategic decision because of the cost and expected lifetime of the units. A line is a route in a railway network that connects two terminal stations, known as origin and destination. A line planning problem (LPP) corresponds to the problem of selecting the set of lines and their frequencies with optimizing two objectives: minimizing the operational cost and maximizing the number of travelers with direct connections (Bussieck et al., 1996; Claessens et al., 1998; Goossens et al., 2004; Caprara et al., 2007). According to Caprara et al. (2007) there are several options in designing a line in order to provide sufficient capacity to transport all passengers: (i) a line that is operated with a high frequency and with low capacity trains; or (ii) a line that is operated with a low frequency and with high capacity trains. The problems investigated in this dissertation are related to and provide results and information related to these options.

2.2.2 Tactical decisions

Tactical decisions are the second level in the railway planning process. These decisions are made more often than strategic decisions. Major problems in this level are timetabling, train routing (i.e. track allocation), rolling stock scheduling, and crew scheduling.

2.2.2.1. Timetabling and scheduling

Train timetabling (i.e. scheduling) is the problem of constructing one or more timetables for trains that have been given a line plan. A timetable, by definition, defines the arrival (departure) time of the train at (from) stations. Train timetables can be divided into two types—those that are cyclic (periodic) and those that are non-cyclic (Caprara er al., 2007; Cacchiani et al, 2010).

Noncyclic train timetables are good for heavy-traffic and long-distance railway networks. Roughly one hundred articles in the literature consider non-cyclic railway timetabling problems. Some of the early efforts in this area include Petersen (1974), Petersen and Taylor (1982), Ceder (1991), Jovanovic and Harker (1991), and Kraay et al. (1991). More recent studies include Carey and Lockwood (1995), Higgins et al. (1996), Brännlund et al. (1998), Caprara et al. (2002), Zhou and Zhong (2005), Caprara et al. (2006), Dessouky et al. (2006), Carey and Crawford (2007), Zhou and Zhong (2007), D'Ariano et al. (2007), Abril et al. (2008b), Castillo et al. (2009), Burdett and Kozan (2009a), Burdett and Kozan (2009b), Liu and Kozan (2009), Burdett and Kozan (2010a), Burdett and Kozan (2010b), Cacchiani et al. (2010), Castillo et al. (2011), Liu and Kozan (2011), Harrod (2011), and Narayanaswami and Rangaraj (2012).

Cyclic train timetables have some advantage over non-cycle ones. The major advantage of cyclic train timetables is that they are easy-to-remember for the passengers. In a cyclic timetable, each trip is operated in a cyclic way, i.e. each period of the timetable is the same. For example, according to this timetable a certain train type *t* heading east always leaves a particular station, say station *s*, at times x:02, x:22, and x:42. The above departure times can be represented as $e + kT$, where e is the original departure time (2 minutes after x in this example), *T* is the cycle time (20 minutes in this example), and *k* is an integer value. We will discuss these two classes of timetables in more details later in this chapter.

2.2.2.2. Track allocation, platforming and train routing

Track allocation and routing problem is the second tactical decision after train timetabling. One important problem in this level is Train Platforming Problem (TPP) which corresponds to routing trains through (and possibly stops at) stations. The problem is very easy to solve for relatively small stations with very small number of alternative path, but for very complex stations, the problem becomes challenging. One possible objective function is the minimization of the sum of delays. The inputs of this problem include the directions of the trains, the scheduled arrival/departure time and complete information about the topology of platforms. It should be considered not only trains that pass the station, but also those trains or locomotives coming (going) from (to) the shunting area. This problem have not received considerable attention from the OR community (Caprara et al., 2007). In a very recent study, Cacchiani et al. (2014a) give an overview of train platforming problem.

2.2.2.3. Rolling stock

The rolling stock scheduling problem corresponds to assign the available stock of locomotives and cars while minimizing the total cost and satisfying all the constraints.

This is important for train operations since the rolling stock acquisition is expensive. In this problem given a planned timetable and the expected number of passengers, the model determines an allocation of rolling stock to the service (Acuna-Agost, 2009). The model can consider the possibility to add or remove cars to trains in some stations.

Moreover it is possible to consider splitting or combining two or more trains. Possible objective functions include: minimization of the expected seat shortages, maximization of a measure of robustness, minimization of the total cost.

This problem first is introduced by Schrijver (1993). The paper considers the problem of minimization the number of train units of different subtypes for an hourly line when satisfying a given seat demand of passengers. Another more complete version of the problem takes into account the changes of the composition of trains with the aim of having an indication of the robustness of the solution. Maroti and Kroon (2007) propose an integer programming model for the problem of the routing units in order to reach regular preventive maintenance. Cordeau et al. (2001) consider a time-space network representing all possible consecutive train sequences that available units can make. The LP relaxation of this model is solved by column generation and integer solutions are then obtained heuristically.

2.2.2.4. Crew scheduling

Crew scheduling is the fourth tactical problem and is defined as finding a set of rosters covering every trip once, satisfying all the operational constraints. A roster is defined as a cyclical sequence of trips performed by each crew. In this problem, a planned timetable for the train services is given that has to be performed every day for a certain period. Also, every service is divided in several trips that are the minimal route served by the same crew. Additionally, every trip starts at a defined time in a depart station and ends in the arrival station. It is necessary that each trip is performed by one crew (Caprara et al., 1999). Crew scheduling problem is a very complex and challenging problem due to both the size of the instances and the type and number of operational constraints (Caprara et al., 2007).

Vaidyanathan et al. (2007) model the crew scheduling problem for North American Railroad by multicomodity network flow and then formulate it as an integer program that can be solved to optimality. In their model, the network flow maps the assignment of crews to train as the flow of crews on the network.

2.2.3. Operational decisions

The third level which corresponds to detailed plan is operational problems. In this level, according to Lusby et al (2011a) all real time decisions that discussed in tactical level are managed in a real-time manner. This class also deals with making decision in real-time where disturbances happen more often due to fluctuation in demand and incidents. Also in some cases, timetables are subject to disturbances. This problem of adjusting existing timetables is referred to as traffic rescheduling, train dispatching and train scheduling under disturbances (Acuna-Agost, 2009).

Railway rescheduling problem (RRP) is one of the most important real-time problems. The recent survey on recovery models and algorithms for train timetable researching, rolling stock rescheduling, and crew rescheduling is done by Cacchiani et al. (2014b). Khosravi et al. (2012) use a modified version of shifting bottleneck heuristic for train rescheduling problem in the UK. Since the original shifting bottleneck procedure has been designed for job shop scheduling problem, they model the train scheduling and rescheduling problem as a job shop scheduling problem. Dündar and Şahin (2013) solve train rescheduling problem by genetic algorithm and artificial neural network. Corman et al. (2011) present an optimization framework for rescheduling trains with different classes of priority. The objective that they consider is delay minimization which is the main objective of the infrastructure manager. This multi-class rescheduling problem is modeled via alternative graph that is solved to optimality by branch-and-bound in each step of the train scheduling problem.

Acuna-Agost (2009) studies train scheduling problem and develops a daily timetable considering some aspects of “detailed platform assignment daily”. Real-time platform assignment appears when a new timetable should be constructed after a disrupted situation, and as a result train should be re-routed at that station. Rodriguez (2007) solves the problem of routing and scheduling at junction in real-time using constraint programming. Recently, a set packing approach has been proposed for real-time junction train routing (Lusby et al., 2013).

The shunting problem can be affected by disturbances as well. Real-time rolling stock problem as a reactive problem deals with the problem of finding a new assignment of train after disturbances. The main objective is to go back as soon as possible to normal operations. The objective of real-time crew scheduling is a combination of feasibility, minimization of operation cost and maximization of stability (Jespersen-Groth et al., 2009). The objective of the crew rescheduling problem is the minimization of the rescheduling costs of crew and the costs of cancelling additional tasks (Cacchiani et al., 2014b).

2.3. Train timetabling and routing with mathematical programs

As this dissertation focuses on the train timetabling and platforming, in this section we present the basic mathematical models presented in the literature for these two classes of NP-hard problems.

2.3.1. Noncyclic train timetabling

In this section the basic non-cyclic timetable is presented that taken from Caprara et al. (2007). A basic non-cyclic train timetable problem consists of a single, one-way line connecting two major stations with a number of stations in between. Let S represent the set of stations, ordered according to the sequence they are appeared along the line for the direction considered, and T denotes the set of trains. Then, a timetable defines, for each train $t \in T$, the departure time from its origin $o_t \in S$, the arrival time at is destination $d_t \in S$, and the arrival and departure times for each intermediate station between origin and destination $o_t < s < d_t$, for all $s \in S$. Each train is then assigned an *ideal timetable* with departure D_{ts} time for each station s: $o_t \leq s < d_t$, and arrival time A_{ts} for each station s: $o_t < s \leq d_t$, which would be the most desirable timetable for the train, that may however be modified in order to satisfy other system constraints such as track capacity which restrict train overtaking. The final solution for the problem will be referred to as the *actual timetable*. The objective is to maximize the sum of the profit of the scheduled trains. The profit achieved for each train depends on the train's *ideal profit* π_t, on the shift v_t, and on the stretch μ_t. Then, the profit for each train t is given by

$$\pi_t - \Phi_t(v_t) - \gamma_t \mu_t \tag{2-1}$$

where $\Phi_t(.)$ is a user-defined non-decreasing function penalizing the train shift, and γ_t is a given nonnegative parameter. Finally, the math model for noncyclic train timetabling can be modeled as follows (Caprara et al., 2002).

Let $G = (V, A)$ be the directed acyclic multigraph in which nodes corresponds to arrival and departures from stations along the line and arcs correspond both to trains stop at a station and to trips from a station s to $s+1$. For each station s, except the first one or origin, defined by U_s is the arrival time, and similarly for each station s, except the last one or destionation, defined by W_s is the departure time. The arc set in this graph is partitioned into arc sets A^t associated with each train $t \in T$. Arcs in A^t from a node $w \in W_{s\text{-}1}$ to a node $u \in U_s$ model train t departing from station s-1 at time instant w and arriving at station s at time instant u. Moreover, arcs in A^t from a node $u \in U_s$ to a node $w \in W_s$ model train t arriving at station s at time instant u and departing from station s at time instant w. To model train flow conservation two artificial nodes are defined: an artificial source node σ, and an artificial sink node τ. The objective function defines the *profit* p_a with each arc $a \in A^t$ for each train t. Before introducing integer linear program, let us first define the decision variable and parameters.

$x_a = \begin{cases} 1 \\ 0 \end{cases}$	If arc a is selected in an optimal solution for all $a \in A^t$ and for all $t \in T$ Otherwise
$\delta^+_t(v)$	The (possible empty) set of arcs in A^t leaving node v, for all $v \in V$, and for all $t \in T$
$\delta^-_t(v)$	The (possible empty) set of arcs in A^t entering node v, for all $v \in V$, and for all $t \in T$
C	The (exponentially large) family of maximal subsets c of pairwise of incompatible arcs

Then, the mathematical program would be

Maximize $\sum_{t \in T} \sum_{a \in A^t} p_a x_a$ (2-2)

Subject to

$$\sum_{a \in \delta_t^+(\sigma)} x_a \le 1 \quad (2\text{-}3)$$

$t \in T$

$$\sum_{a \in \delta_t^-(v)} x_a = \sum_{a \in \delta_t^+(v)} x_a \qquad t \in T, v \in V \backslash \{\sigma, \tau\} \qquad (2\text{-}4)$$

$$\sum_{a \in c} x_a \le 1 \qquad c \in C \qquad (2\text{-}5)$$

$$x_a \in \{0,1\}, \quad a \in A \qquad (2\text{-}6)$$

The objective function (2-2) is the sum of the profits of the arcs related to each path in the solution. Constraint set (2-3) imposes that at most one arc associated with a train can be selected to leave the source node σ. Constraint set (2-4) is the flow conservation associated with number of arcs entering and leaving each arrival or departure node. Finally, constraint set (2-5) is a clique constraint and forbids the simultaneous selection of incompatible arcs. For a recent survey on non-periodic train timetabling see Cacchiani et al. (2014a).

2.3.2. Cyclic train timetabling

Serafini and Ukovich (1989) are the first ones who developed a model for generating cyclic timetable. The model is called Periodic Event Scheduling Problem (PESP). Harrod (2012) mentions that the Periodic Event Scheduling Problem (PESP) provides the modeling framework for the vast majority of cyclic timetables (Kroon and Peeters, 2003; Liebchen, 2008). The PESP considers the problem of the scheduling a set of periodically recurrent events under time-windows constraints. In the PESP, all calculations are carried out modulo T, where T is the cycle time. A typical constraint in the PESP is the requirement that a certain process, that starts and ends at event times v_e and $v_{e'}$ respectively, has a duration of at least $L_{e,e'}$ but no more than $U_{e,e'}$ ($0 \le L_{e,e'} \le U_{e,e'} < T$).

This constraint is modeled by the inequality $L_{e,e'} \leq (v_{e'} \_ v_e) \bmod T \leq U_{e,e'}$ where $0 \leq L_{e,e'}$, $v_{e'}$, v_e, $U_{e,e'} < T$. The nonlinear modulus operator is then removed by introducing a binary decision variable $Q_{e,e'}$ that indicates whether the process crosses the end of the cycle (=1) or not (=0). Everything is computed mod T, so $v_{e'}$ is less (greater) than v_e if the process crosses (does not cross) the end of the cycle. The final constraint—$L_{e,e'} \leq (v_{e'} \_ v_e) + T \times Q_{e,e'} \leq U_{e,e'}$—is linear and fits within a standard mixed integer programming (MIP) framework (Kroon et al., 2008). According to Caprara et al. (2007), variables $Q_{e,e'}$ make the PESP quite hard to solve by standard branch-and-bound methods, due to the relatively large coefficients of T and the LP relaxations of models based on this formulation of PESP are quite weak. Other possibilities for the PESP are to consider L and U as time instants instead of durations and/or to allow the event times v_e and $v_{e'}$ to be any real number ≥ 0 and $Q_{e,e'}$ to be integral.

The PESP is the basis for most cyclic railway timetabling studies including the works by Odijk (1996), Peeters and Kroon (2001), Kroon and Peeters (2003), Lindner and Zimmermann (2005), Liebchen (2008), Liebchen and Möhring (2008), and Liebchen, et al. (2008). Nachtigall and Voget (1996) perform a study outside the PESP paradigm in which genetic algorithms are used to optimally synchronize the operations on two or more railway lines that operate with different frequencies.

Other approaches for modeling cyclic timetable are graph theory and machine scheduling problem. Caprara et al., (2001, 2002) build a periodic timetable in a unidirectional track by representing the problem using a directed graph. The graph is then transformed to an integer programming model that is solved by a Lagrangian relaxation approach.

Surprisingly, to our knowledge, Bergmann (1975) remains the only article to consider a cyclic train timetabling problem in which the minimization of the cycle length is the primary objective. The minimization of the cycle length requires the cycle time T to be a decision variable. However, within the PESP paradigm, this leads to a quadratically constrained formulation in which the two decision variables T and $Q_{e,e'}$ are multiplied together (see above). By avoiding the PESP framework, Bergmann (1975) manages to create a linear formulation for a simple periodic timetabling problem in which the cycle time T is a decision variable and also the objective to be minimized.

2.3.3. Solution approaches to train scheduling and timetabling

Regardless of the nature of train timetable and modeling techniques that have been used, researches applied many techniques to solve the train timetabling and scheduling problem which its integer program formulation is known to be NP-hard (Cai and Goh, 1994). In this section, we briefly review the efforts that the researches have been made for solving this class of NP-hard problems. Particularly, we look at this problem from modeling method and solution approaches.

De Oliveira (2001) considers the cyclic timetabling problem as a special case of the job-shop scheduling problem where jobs are trains and machines are track (sections) and solves it by constraint programming. Other works that model train scheduling problem as a job shop problem include Liu and Kozan (2009), Burdett and Kozan (2009c), Burdett and Kozan (2010b), Khosravi et al. (2012). Other approaches include lagrangian (Cacchiani et al., 2012; Caprara et al., 2006; Brännlund et al., 1998), branch and bound (D'Ariano et al., 2007), column generation (Min et al., 2011), constraint generation (Odijk, 1996), Genetic Algorithm (Nachtigall and Voget, 1996; Chung et al., 2009), decomposition algorithm (Peng et al., 2013), modulo simplex algorithm (Goerigk and Schobel, 2013), Hypergraph (Harrod, 2011), independent set problem (Caimi et al., 2009), and max-plus algebra (Goverde, 2007 and 2010).

2.3.4. Train platforming and routing

Billionnet (2003) presents one of the first advances on this topic after the original work of (DeLuca-Cardillo and Moine, 1998). They both formulate the problem as a graph-coloring problem which is based on a node packing formulation. We present here the model developed by Billionnet (2003). For this purpose, let us define the binary decision variable x_{ir}, equal to 1 if vertex v_r (r = 1, …, n) is assigned color $i \in L(V_r)$, and then formulate the k L-list τ graph coloring as the following feasibility problem:

Find x_{ir} (2-7)

Such that

$$\sum_{i \in L(V_r)} x_{ir} = 1 \qquad r = 1, \ldots, n \qquad (2\text{-}8)$$

$$x_{ir} + x_{is} \leq 1 \qquad r < s,\ \{V_r, V_s\} \in E,\ i \in L(V_r) \cap L(V_s) \qquad (2\text{-}9)$$

$$x_{ir} + x_{js} \leq 1 \qquad \{V_r, V_s, i, j\} \in \tau \qquad (2\text{-}10)$$

$$x_{ir} \in \{0, 1\} \qquad r = 1, \ldots, n,\ i \in L(V_r) \qquad (2\text{-}11)$$

The binary variables determine the assignment of a particular platform to a given train. Therefore, by considering this type of decision variable any objective function would seek to maximize (minimize) the use of a certain platform within a station (Lusby et al., 2011a). In the above formulation, constraint (2-8) enforces each train to be assigned to a platform. Two train should not and cannot be assigned to the same platform at the same time (constraint (2-9)), and according to (2-10) two trains cannot be assigned different platform if their respective paths to the same platform conflicts. Other studies consider and model this problem as a node packing problem solved by a branch and cut approach (Zwaneveld et al., 1996) and as a circle graph, permutation graph, graph coloring and their combination (Demange et al., 2012).

2. 4. Line capacity

The current study also has significant overlap with the subject of railway capacity. Capacity is extremely dependent on infrastructure, traffic and operating parameters (Abril et al., 2008a). Railway capacity is normally defined as the maximum number of trains that can traverse a given section of a track in a given duration of time. Capacity depends on the *particular mix of the trains and the order in which they run over line*. Recent studies that focus on railway capacity issues include Burdett and Kozan (2006), Abril et al. (2008a), Dingler et al. (2009), Harrod (2009), and Salido and Barber (2009). Indeed, all of the above articles adopt this definition of railway capacity.

Surprisingly, to our knowledge, no article besides Bergmann (1975) has focused on the alternate definition used in the current study in which capacity is *the minimum cycle length that can feasibly accommodate a given number of trains over a given section of track in each cycle*. Note that this alternate definition of capacity is directly related to the standard definition. Indeed, by squeezing a given number of trains into a smaller cycle length, we are increasing the number of trains that can be squeezed into a given duration of time.

Burdett and Kozan (2006), Abril et al. (2008a), Dingler et al. (2009), Harrod (2009) consider capacity in the context of a non-cyclic timetable. The article by Salido and

Barber (2009) is the study most closely resembling the current investigation in that it considers capacity in the context of a cyclic timetable and considers different values for the length of the cycle (e.g. 140, 120, 100, 90, 75, 60 min). However, Salido and Barber (2009) mainly focus on the standard definition of railway capacity, and they do not present a mathematical program whose objective is to minimize the length of the cycle.

2. 5. Objectives of mathematical programs for railway timetabling and routing

The objectives considered in train related articles include minimizing total time used by the passengers in the system (including travel time and waiting time), minimizing passenger waiting time (Lindner and Zimmermann, 2005), minimizing makespan (Burdett and Kozan, 2010a), minimizing the maximum duration between consecutive train arrivals at a station (Ceder, 1991), minimizing the cost of train delay and energy consumption (Kraay et al., 1991), minimizing the number of trains (Peeters and Kroon, 2001), minimizing the delays of trains at destination and train operating cost (Higgins et al., 1996), minimizing the total cost of train stopping and waiting times at sidings (Cai and Goh, 1994), and maximizing profit (Caprara et al., 2002). The minimization of traveling time may arise in network timetabling or in case where train speed is not fixed between two connections and/or stations. Train schedule reliability is also considered as an objective to be maximized for single track rail line (Ferreira and Higgins, 1996). For measuring reliability Ferreira and Higgins (1996) use the amount of risk of delay (RD) associated with a given schedule as the reliability component of the optimization model.

In summary, a detailed examination of the literature has yielded scores of outstanding contributions in the areas of train timetabling, train platforming, and railway capacity analysis. According to Gorman (2010), train scheduling is the most popular topic in the railway operations literature, comprising about 19% of all articles published. Furthermore, the general topic of train scheduling and track capacity accounts for about 40% of all articles in the railway operations literature that present optimization models. Thus, the general subject of the current investigation is hardly unfamiliar. However, the current study appears to be unique in many ways. First, it is the only study—after Bergmann (1975) —to present an MILP model for a cyclic train timetabling problem in which the length of the cycle time is a decision variable. Second, it is the only study—after Bergmann (1975)—to present an MILP model for a cyclic train timetabling problem where the primary objective is to minimize cycle length. Third, it is the only study that adopts the Bergmann's definition of capacity, which is as the minimum cycle length that can feasibly accommodate a given number of trains over a given section of track. Therefore, in our approach capacity not only considers the track restriction in each cycle, but also integrates the station capacity. Fourth, it presents the literature's first MILP model for a cyclic, combined train timetabling and platforming problem for single track unidirectional and bidirectional railway. Fifth, it presents the literature's first MILP model for a cyclic train platforming problem in which the length of the cycle is a decision variable. Sixth, it is the first study to present a MILP model for a cyclic train platforming problem where the primary objective is to minimize cycle length. Seventh, this dissertation considers any number of train types per cycle. Eighth, we allow stations to have more than one siding. Finally, we allow trains to start or end at intermediate stations.

Chapter 3

Mixed integer programming for minimizing the period of a cyclic timetable for a single track with two train types

3.1. Introduction and problem description

The material in this chapter has been published in Heydar et al. (2013). This work extends the work of Bergman (1975) to investigate the capacity of a single track, unidirectional rail line that adheres to a cyclic timetable. This problem is a basic problem in domain of train scheduling and timetabling. A set of intermediate stations lies between an origin and destination with one siding at each station. Two types of trains–express and local–are dispatched from the origin in alternating fashion. The local stops at every intermediate station and the express stops at no intermediate stations. Constraints include a minimum stopping (dwell, delay) time for the local train at each station, a maximum total dwell time for the local train at all stations combined, and headway considerations on the main line and in stations. Bergmann (1975) develops a mixed integer linear program for obtaining a feasible timetable with the objective of minimizing the length of the dispatching cycle. However, no numerical study is presented. In this chapter, we restate the problem from Bergmann (1975) using improved notations; modify Bergmann's mathematical model by adding a second objective and removing unnecessary variables; and perform the first numerical analysis of this problem by considering hundreds of randomly generated instances with up to 70 stations.

Topological configuration of this railway system is depicted in Figure 3.1. In this figure the paths of express and local trains are shown by dashed and solid arrows above and below the track, respectively. Consider a situation in which trains are dispatched at regular intervals on a single, unidirectional track from an origin to a destination. Two types (categories) of trains are dispatched. The first category consists of local trains that stop at all S intermediate stations lying between the origin and destination. The second category consists of express trains that do not stop at any of the intermediate stations. No passing is allowed on the main track. However, all stations in the system are located in the siding, so it is possible for an express train to pass a local train at a station. Thus, an express train's passage of a station being served by a local train is unobstructed. Further each station siding is sufficiently long so that deceleration and acceleration by a local train moving into or out of a station does not interfere with the operations of the express trains in the traffic stream which the local train is leaving or entering.

Local and express trains are dispatched alternately in order to create a cyclic timetable in which one train of each type is dispatched per cycle and the cycle length, *Interval*, is a decision variable. In particular, local trains are dispatched from the origin at times $i \times Interval$, $i \in \{0, 1, 2, \ldots\}$, and express trains are dispatched from the origin at times $i \times Interval + \Delta$, $i \in \{0, 1, 2, \ldots\}$. The decision variable Δ can have any value such that $0 < \Delta < Interval$ and all other constraints described below are satisfied. From this definition it is revealed that the tth local train leaves the origin at time $(t - 1) \times Interval$, and the tth express train leaves the origin at time $(t - 1) \times Interval + \Delta$. The cyclic nature of the timetable means that all local trains are identical and all express trains are identical except for their departure times from the origin.

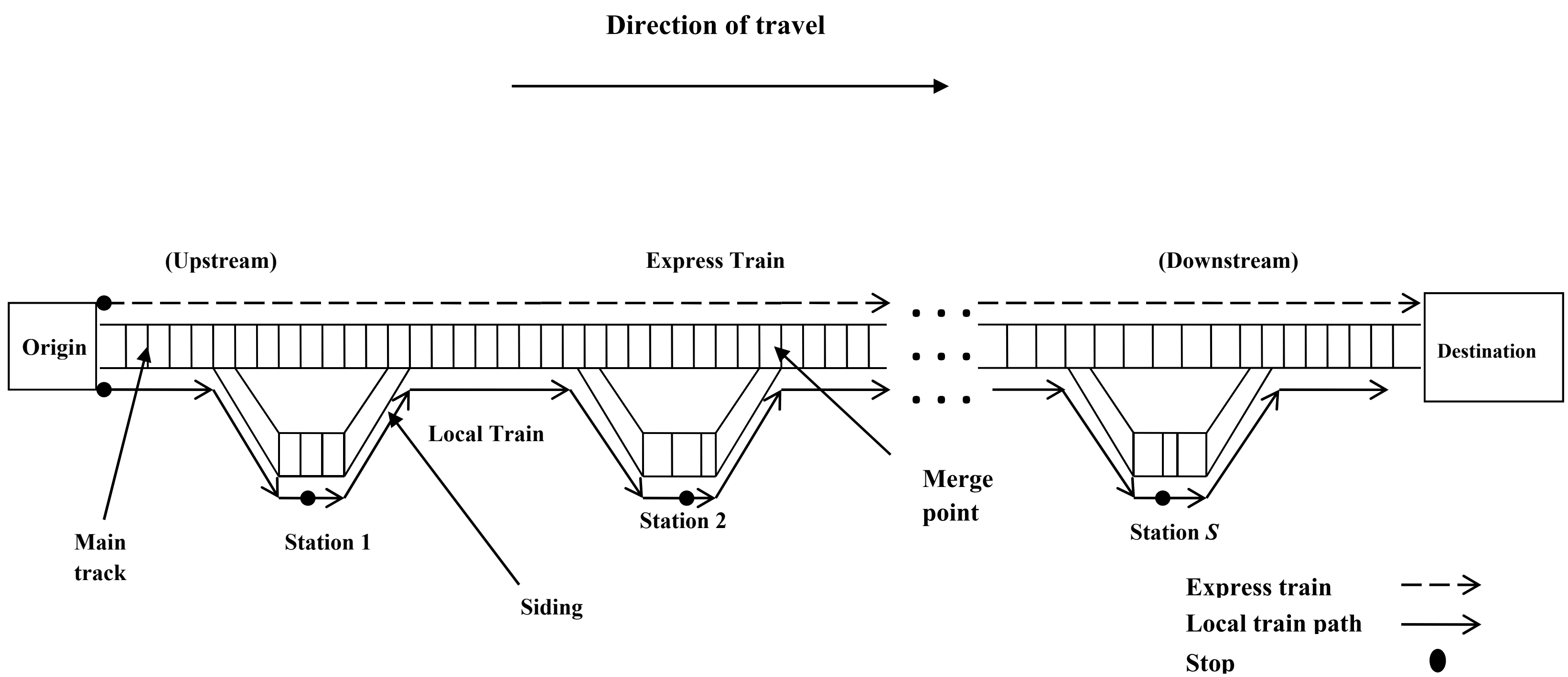


Figure 3.1 Topology of railway system investigated in this study

All trains travel at the same speed while on the main track. Let $trav_s$ be the time required by a train making no stop to travel from the origin to the merge point–the point at which local trains return to the main track–just downstream from station s. Denote by $dMin_s$ the minimum required delay or increase in train travel time due to stopping at station s rather than passing it. This is the minimum amount of time that a train must spend decelerating, unloading passenger, loading passengers, and accelerating on the siding beside station s in order to serve station s. For simplicity, we refer to this term as the minimum stopping (delay, dwell) time of a local train at station s. Denote by D_s the actual stopping (delay, dwell) time of a local train at station s. D_s is a decision variable that must be greater than or equal to $dMin_s$. A local train dispatched from the origin at time 0 reaches the merge point just downstream from station s at time $\left[trav_s + \sum_{q=1}^{s} D_q \right]$.

In this study two objectives are considered. The primary objective is to minimize *Interval* subject to four types of constraints. First, the actual dwell time D_s of a local train in station s must be greater than or equal to the minimum required dwell time $dMin_s$. Second, the total dwell time of the local train at all stations combined may not exceed *dMax*. Third, trains on the main track must be separated by a minimum headway of *hTrack* minutes. Finally, the departure time of a train from station s and arrival time of the next train after it into station s must be separated by a minimum headway of $hStation_s$ minutes. For simplicity, we assume that a train's "departure time" from a station occurs at the very end of its dwell time D_s in the station, and we assume that a train's "arrival time" into a station occurs at the very beginning of its dwell time D_s in the station. The secondary objective is to minimize the total delay $\sum_{s=1}^{S} D_s$ experienced by the local train at all intermediate stations combined.

The primary objective directly relates to track capacity. Indeed, in order to determine the maximum capacity of a single track, we can either determine the maximum number of trains that can be dispatched within a period (or "cycle") of given length, or we can determine the minimum cycle length such that a given number of trains can feasibly dispatched within one cycle. While almost all articles in the literature investigate capacity within the former conceptual paradigm, the study in this dissertation is unique in that it presents MILP models for investigating track capacity within latter paradigm.

Figure 3.2 presents a time-space diagram of the type of cyclic train timetable described above. Each local (express) train is depicted using a solid (dashed) line. The slopes of all lines between stations are equal because all trains move at the same speed on the main line. Note that all terms described in the preceding paragraphs–including *Interval*, the length of the dispatching cycle–appear in the figure. The local train that departs the origin at time 0 is not depicted in order to create space for these terms.

We now make a few additional comments and assumptions regarding the problem. First, we assume there is no limit on the number of trains available. That is, rolling stock components (i.e. locomotives, railcars, train sets) are assumed to be available whenever and wherever they are needed. Second, we assume the origin and destination have unlimited capacity to accommodate trains. In other words, we ignore any constraints on the operations at and before the origin, and at and after the destination. However, the headway constraints on the main track immediately after the origin and immediately before the destination are considered. Note that the origin and destination have been ambiguously defined so either or both of these locations may represent stations of simply points on a track. However, the destination is also known as “stations $S + 1$” so that the parameter $trav_{S+1}$ can be used do denote the travel time from origin to destination.

In this chapter, the problem from Bergmann (1975) is restated using the new notations introduced above; mathematical model is modified by adding a second objective and removing unnecessary variables; the first numerical analysis is performed by considering hundreds of randomly generated instances with up to 70 stations; and an algorithm is proposed to identify total number of optimal solutions.

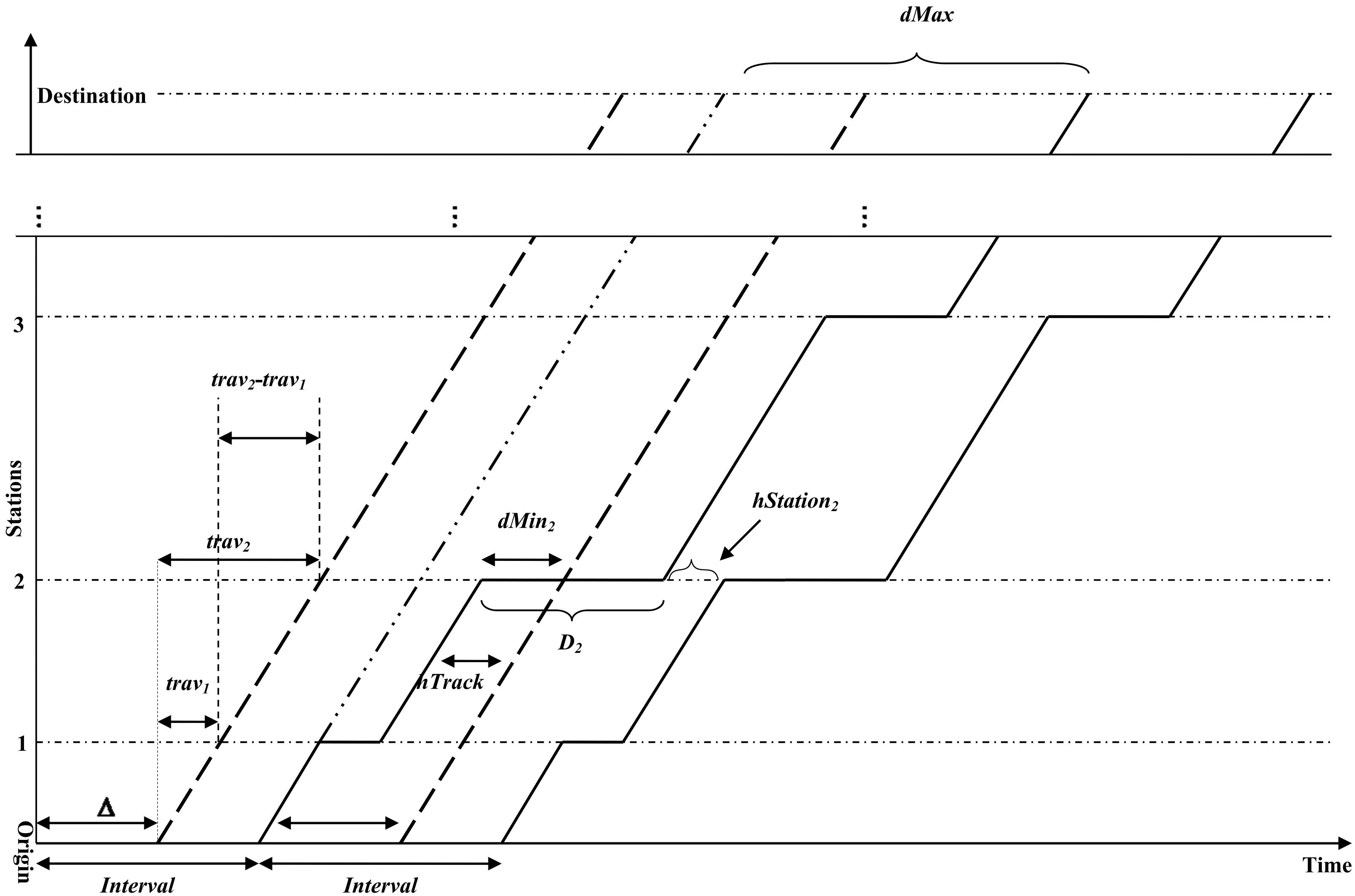


Figure 3.2 Cyclic train timetable depicted as a time-space diagram

3.2. Review of related works

This chapter focuses on passenger railway systems. Within this category, we focus on timetable optimization—otherwise known as train scheduling. As discussed in Chapter 1 (Figure 1.1), timetabling is one of the four main tactical problems that are solved during the overall railway planning process. As mentioned in Chapter 1 and according to Gorman (2010), train scheduling is the most popular topic in the railway operations literature, comprising about 19% of all articles published. Furthermore, the general topic of train scheduling and track capacity accounts for about 40% of all articles in the railway operations literature that present optimization models. Thus, the general subject of the current investigation is hardly unfamiliar.

Some of the early efforts in non-cyclic include Petersen (1974), Petersen and Taylor (1982), Ceder (1991), Jovanovic and Harker (1991), Kraay et al. (1991), Carey and Lockwood (1995), Higgins et al. (1996), Brännlund et al. (1998), Caprara et al. (2002), Zhou and Zhong (2005), Caprara et al. (2006), Dessouky et al. (2006), Carey and Crawford (2007), Zhou and Zhong (2007), D'Ariano et al. (2007), Abril et al. (2008b), Castillo et al. (2009), Burdett and Kozan (2009a), Burdett and Kozan (2009b), Liu and Kozan (2009), Burdett and Kozan (2010a), Burdett and Kozan (2010b), Cacchiani et al. (2010), Castillo et al. (2011), Liu and Kozan (2011), Harrod (2011), and Narayanaswami and Rangaraj (2013).

We have already shown that cyclic railway timetabling has been addressed by about 30 articles in the literature which most of them are based on the concept of Periodic Event Scheduling Problem (PESP) first introduced by Serafini and Ukoich (1989) in their seminal paper. The PESP is known as an NP-hard problem (Caprara et al., 2007). Those studies that have been carried out based on the PESP include Odijk (1996), Nachtigall and Voget (1996), Peeters and Kroon (2001), Kroon and Peeters (2003), Lindner and Zimmermann (2005), Liebchen (2008), Liebchen and Möhring (2008), Liebchen et al. (2008), and Siebert and Goerigk (2013). Bergmann (1975) remains the only article that model cyclic train timetabling by mixed integer programming approach with the cycle length as the primary objective and decision variable which will create a quadratically constrained problem in the PESP paradigm.

The current study gives an alternate definition of railway capacity and provides another method for finding the optimal level of line capacity by minimizing the cycle length. Recent studies that focus on railway capacity include Burdett and Kozan (2006), Abril et al. (2008a), Dingler et al. (2009), Harrod (2009), and Salido and Barber (2009). To the best of our knowledge, no article besides Bergmann (1975) has focused on the alternate definition used in the current study.

Regarding the objective function, the work by Bergmann (1975) appears to be unique in that it is the only article to (1) consider a cyclic timetabling problem in which the minimization of the cycle length is the primary objective; (2) present a linear formulation of a cyclic timetabling problem in which the length of the cycle is a decision variable; and (3) focus on the alternate definition of railway capacity in which capacity is the minimum cycle length that can feasibly accommodate a given number of trains over a given section of track in each cycle. According to a Google Scholar, the article by Bergmann (1975) has not been cited even once as of May 31, 2013. This helps to explain why it remains a unique contribution in the literature despite being dated by almost 40 years. One shortcoming of Bergmann (1975) is that the math model lacks an accompanying numerical investigation. In the current study, we present a slightly modified version of the mathematical formulation presented in Bergmann (1975) and perform the first numerical study of the problem in which hundreds of problem instances with up to 70 stations are solved to optimality within a reasonable amount of computation time.

3.3. Mathematical formulation

In this section the mixed integer programming model is presented. The set of indices, parameters and decision variables, and their respective explanation, used to define the mathematical program are given in Table 3.1. The input data consists of eight primary parameters (S, $trav_s$, $dMin_s$, $hTrack$, $hStation_s$, $dMax$, w_k and M) and one secondary parameter *maxExpr* that is derived from the primary parameters. The primary parameters are described in Section 3.1. For this model four decision variables are defined that will be defined in this section.

In this model, three decision variables are related to time; hence they take real values. A by-product of this model is to sequence trains on the main track when a local train join the main track at the merge point just downstream any station. In other words, we must decide either local or express train goes first. The four decision variables are *Interval*, D_s, Δ, and the binary variable Y_{ts}.

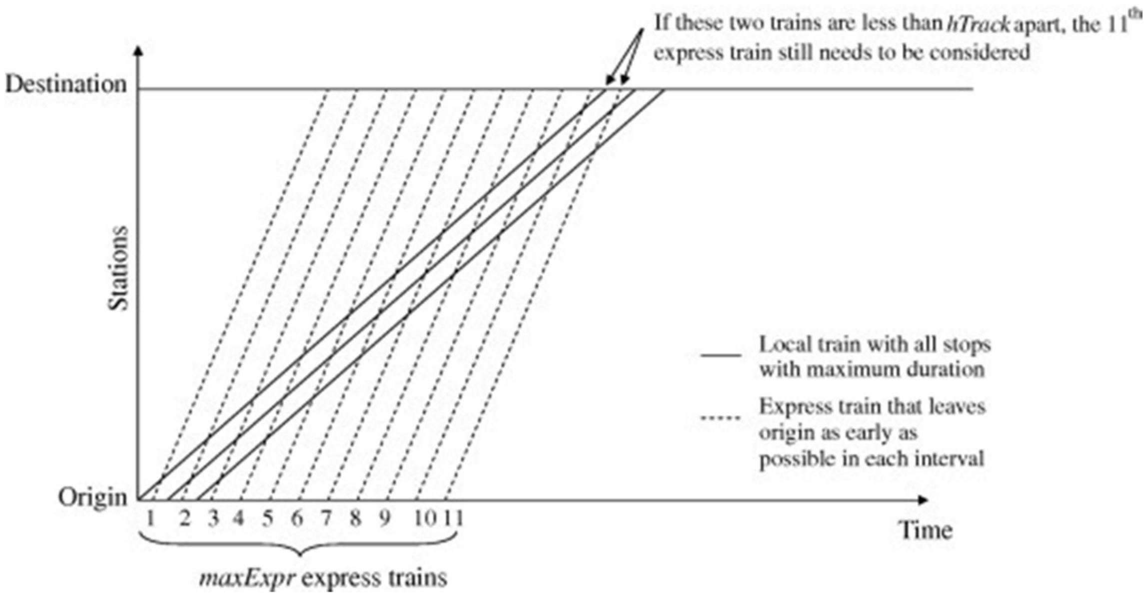


Figure 3.3 Illustration of the number of express trains (*maxExpr* = 11 in this case) that must be compared to the local train that departs the origin at time 0 in order to guarantee solution feasibility with respect to the headway *hTrack* on the main line.

The secondary parameter, *maxExpr*, represents the maximum number of express trains that could possibly come within *hTrack* minutes of the first local train (i.e. the local train that departs at time 0) on any portion of the main track. As illustrated in Figure 3.3, *maxExpr* gives the number of express trains that must be compared to first local train in order to guarantee solution feasibility with respect to the headway *hTrack* on the main line for the first local train versus all express trains. But, owing to the repetitive, cyclic nature of the timetable, the satisfaction of all main track headway constraints involving the first local train and all express trains implies satisfaction lf all main track headway constraints involving all local trains and all express trains. Note that *maxExpr* also provides an upper bound on the number of express trains that each local train meets during its journey. This latter quantity also equals the number of local trains that each express train passes during its journey.

Table 3.1 Indices, parameters and decision variables in mathematical model

Indices	
k	Objective function component ($k = 1, 2$)
s	Station ($1 \le s \le S$).
t	Express train
Parameters	
S	Number of intermediate stations excluding the origin and destination stations (integer, > 0).
$trav_s$	Time required in minutes for an express train to travel from origin to the point at which local train returns to the main track just downstream from station s (real, > 0, $1 \le s \le S+1$).
$dMin_s$	Minimum dwell time in minutes at station s for a local train (real, > 0, $1 \le s \le S$)
$hTrack$	Minimum headway in minutes between successive trains on main track (real, > 0).
$hStation_s$	Minimum required separation in minutes between two consecutive trains arriving at station s. This parameter combines the station approach velocity, emergency breaking system response time, train length, and level of operational deceleration, emergency deceleration and acceleration (real, > 0, $1 \le s \le S$).
$dMax$	Maximum allowed difference in time (in minutes) which local and express trains require to travel between origin and destination (real, > 0).
w_k	Weight of objective k in the objective function (real, $0 \le w_k \le 1$).
M	A large positive number.
$maxExpr$	Maximum number of express trains that could possibly come within $hTrack$ minutes of the first local train on the main track.
Decision Variables	
$Interval$	Interval at which both local and express trains are dispatched (real, > 0).
D_s	Dwell time for a local train at station s or increase in train travel time due to stopping at station s rather than passing it (real, > 0).
Δ	Time when the first express train departs from the origin (real, > 0).
$Y_{st} = \begin{cases} 1 \\ 0 \end{cases}$	If express train t arrives at the merge point just downstream from station s before the local train departing from the origin at time 0 arrives there. Otherwise (binary)

The computation of $maxExpr$ proceeds as follows. First, to maximize the number of express trains that could interfere with the first local train, we assume the first local train makes its journey. Using the maximum allowed time $trav_{S+1} + dMax$ from origin to destination, it therefore reaches the destination at time $trav_{S+1} + dMax$. Then, to maximize the number of express trains that could interfere with the first local train, we assume that each express train leaves the origin as early as possible in each interval. So we assume the tth express train leaves the origin at time $(t - 1) \times Interval + hTrack$ and reaches the destination at time $trav_{S+1} + (t - 1) \times Interval + hTrack$. The secondary parameter $maxExpr$ is the largest possible value of t such that the tth express train could reach the destination less than *hTrack* minutes after the first local train reaches there. In other words, *maxExpr* is the largest integer t such that

$$trav_{S+1} + (t - 1) \times Interval + hTrack < (trav_{S+1} + dMax) + hTrack \tag{3-1}$$

This can be simplified to

$$(t - 1) \times Interval < dMax \tag{3-2}$$

and further expressed as

$$t < \frac{dMax}{Interval} + 1 \tag{3-3}$$

The unknown *Interval* appears in the denominator of (3-3). We therefore replace it with the higher of its two known lower bounds–$2 \times hTrack$ and $max_s \{dMin_s + hStation_s\}$–that are explained by the mathematical program below. In particular, the lower bound $2 \times hTrack$ is identified by summing constraints (3-10) and (3-11), and lower bound $max_s \{dMin_s + hStation_s\}$ is established by logically combining constraints (3-9) and (3-7). The replacement of *Interval* with its lower bounds ensures that *maxExpr* is conservatively calculated. In other words, *maxExpr* equals the highest value that is theoretically possible based on the extreme case where the length of each interval is as small as possible. The parameter *maxExpr* is therefore the largest integer t such that

$$t < \frac{dMax}{max\left\{2 \times hTrack, max_s\left\{dMin_s, hStation_s\right\}\right\}} + 1 \tag{3-4}$$

Note the strict inequality above. It follows that

$$maxExpr = \left\lceil \frac{dMax}{max\left\{2 \times hTrack, max_s\left\{dMin_s, hStation_s\right\}\right\}} \right\rceil \tag{3-5}$$

In the above manner, *maxExpr* can be calculated based on the other input parameters and determined before the constraints in the math model are constructed.

The four decision variables in the model are *Interval*, D_s, Δ, and Y_{st}. As mentioned earlier, the first three variables take real values, and the last variable is binary. As mentioned in Section 3.1, *Interval* is the length in minutes of the dispatching cycle. It is the quantity we seek to minimize in this research. D_s is the stopping (delay, dwell) time of each local train in station *s*, and Δ is the time when the first express train departs the origin. The binary Y_{st} variables indicate the sequence of trains on the main line; they help to enforce the headway restrictions on the main line.

The mixed integer programming formulation of this problem is as follows:

$$\text{Minimize } w_1 \times Interval + w_2 \times \sum_{s=1}^{S} D_s \tag{3-6}$$

Subject to

$$D_s \geq dMin_s \qquad \forall s: 1 \leq s \leq S \tag{3-7}$$

$$\sum_{s=1}^{S} D_s \leq dMax \tag{3-8}$$

$$Interval \geq D_s + hStation_s \qquad \forall s: 1 \leq s \leq S \tag{3-9}$$

$$\Delta \geq hTrack \tag{3-10}$$

$$Interval - \Delta \geq hTrack \tag{3-11}$$

$$\left[(t-1)Interval + \Delta\right] - \sum_{q=1}^{s} D_q + MY_{st} \geq hTrack$$

$$\forall s: 1 \leq s \leq S \quad \forall t: 1 \leq t \leq maxExpr \tag{3-12}$$

$$\left[(t-1)Interval + \Delta\right] - \sum_{q=1}^{s} D_q - M\left(1 - Y_{st}\right) \leq -hTrack$$

$$\forall s: 1 \le s \le S \quad \forall t: 1 \le t \le maxExpr \tag{3-13}$$

The objective function (3-6) has two parts, weighted w_1 and w_2, which respectively pursue the minimization of the dispatching cycle and the minimization of the total delay experienced by the local train at all stations combined. Various objectives may be considered by changing the values of weights w_1 and w_2. The first objective is the main focus of this research, so $w_1 >> w_2$ in the most experiments in this research. Constraint (3-7) defines the minimum dwell times for local trains at each station. Variable D_s has some minimum value, $dMin_s$, that is unique for each station. Constraint (3-8) ensures that the local train's total dwell time in all stations combined does not exceed the maximum allowed value *dMax*. The main goal of this constraint is to measure the service level provided by the railway system. In other words, by putting an upper bound on the sum of dwell times the model tries to minimize travel time once a local train is dispatched from the origin. Moreover, this constraint represents the maximum difference in time which local and express trains require in traveling from stations 0 to station S (origin to destination). Constraint (3-9) enforces a minimum headway between trains in the stations. In particular, it ensures that the cycle length *Interval* is large enough so that the local train's stop in each station s can be made without violating the minimum station headway value $hStation_s$. In other words, the departure time of a local train from station s and arrival time of the next local train after it into station s must be separated by a minimum headway of $hStation_s$ minutes. Constraints (3-10) and (3-11) enforce the headway restriction on the main line between the origin and station 1.

The last two constraints (3-12) and (3-13) are disjunctive constraints that enforce the headway restrictions on the main line between station 1 and the destination. In particular, these constraints guarantee that express train t appears at the merge point just downstream from station s either *hTrack* minutes after (constraint (3-12)) or *hTrack* minutes before (constraint (3-13)) the first local train appears there for all t from 1 to *maxExpr* and for all s from 1 to S. In these constraints, the value $trav_s$ has been removed from the traveling times of both express train t and the first local train because the $trav_s$ values cancel each other out when the latter quantity $\left[trav_s + \sum_{q=1}^{s} D_q\right]$ is subtracted from the former $[trav_s + (t - 1) \times Interval + \Delta]$. Note that, as argued earlier in this section, although constraints (3-12) and (3-13) only consider one local train and the first *maxExpr* express trains, it enforces headway constraints on the main track for all trains and all express trains owing to the repetitive, cyclic nature of the timetable.

3.4. Illustrative examples

In this section two small examples are given to depict how this model works. In the next section experimental study will be done. The above mathematical formulation was coded into Microsoft Visual C++ 2010. The code includes a procedure for automatically computing *maxExpr* before the constraints are constructed. ILOG Concert Technology was used to define the model within C++ and call the mixed integer linear programming solver IBM ILOG CPLEX 11.2 to solve instance within Windows XP environment on an IBM-compatible desktop computer with a 2.0 GHz processor and 2 GB of RAM.

3.4.1 Illustrative example 1:

For purpose of illustration a typical small size problem is solved. The input data for this problem instance is given in Table 3.2. The result of this example in a form of timetable is given in Table 2.3. The timetable displays the values of *Interval* and D_s at the top and detailed schedules for the first five local and five express trains at the bottom. The schedule of each train is fully defined by the arrival and departure times of the train at each station as shown in Table 3.3. This solution is obtained in less than a second. Figure 3.4 shows the same solution displayed in the form of time-space diagram. The diagram displays the progression of the first three local trains and first four express trains from the origin to destination. Local (express) trains are indicated with solid (dashed) lines.

As Table 3.3 and Figure 3.4 indicate, the optimal value for this instance–the minimum value of *Interval*–is 4 min. This value is higher than the obvious lower bound for *Interval* owing to station headway constrains alone ($max_s\{dMin_s + hStation_s\} = 2$), and it is also higher than the obvious lower bound for *Interval* owing to main track heady constraints alone ($2\times hTrack = 3$). In addition, the minimum value of *Interval* is also higher than the obvious lower bound of 3.5 that can be obtained by considering the main line headway constraints before and after, and the station headway constraints within, a single station where an express train passes a local train. Thus, the cyclic nature of the timetable and inclusion multiple stations lead to a non-intuitive result even for this small problem instance.

Table 3.2 Input data for example 1

S	$hTrack$	$hStation_s$	$dMax$	w_1	w_2
4	1.5	0.5 for all s	15	1	0
	Station 1	Station 2	Station 3	Station 4	Destination
$dMin_s$	0.5	0.5	1.5	1.5	-
$trav_s$	1.5	5	10.5	14	15
$maxExpr$	5				

Note in Table 3.3 and Figure 3.4 that the total delay time for each local train is 3.5+0.5+3.5+3.5=11 min which is less than the maximum allowed delay of 15 min. In addition, note that each local train is passed by three express trains–in stations 1, 3, and 4– and each express train passes three local trains–also in stations 1, 3, and 4. Also, the actual number of express trains that come within *hTrack* = 1.5 min of the first local train on any portion of the main track is 3. However, a conservative *maxExpr* = 5 express trains are compared to the first local train in constraints (12) and (13) to guarantee, beyond a doubt, that no express train are ever less than the minimum headway *hTrack* minutes away from the first local train on the main line.

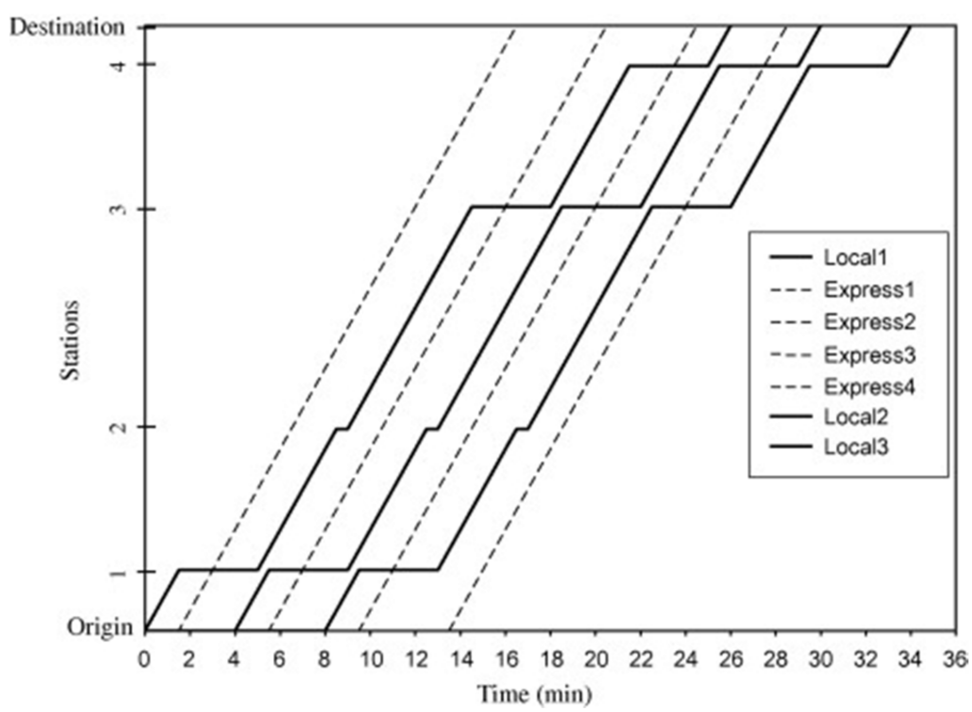


Figure 3.4 Time-space diagram for the first optimal timetable in Example 1.

Table 3.3 An optimal timetable for example 1 (passing occurs in stations 1, 3, 4)

Minimum *Interval*		4				
		Station 1	Station 2	Station 3	Station 4	Destination
D_s		3.5	0.5	3.5	3.5	-
	Departure time from origin					
Local	0	1.5-5	8.5-9	14.5-18	21.5-25	26
Express	1.5	3	6.5	12	15.5	16.5
Local	4	5.5-9	12.5-13	18.5-22	25.5-29	30
Express	5.5	7	10.5	16	19.5	20.5
Local	8	9.5-13	16.5-17	22.5-26	29.5-33	34
Express	9.5	11	14.5	20	23.5	24.5
Local	12	13.5-17	20.5-21	26.5-30	33.5-37	38
Express	13.5	15	18.5	24	27.5	28.5
Local	16	17.5-21	24.5-25	30.5-34	37.5-41	42
Express	17.5	19	22.5	28	31.5	32.5

It is easy to see that there are multiple optimal solutions for this problem instance. Another optimal solution can be obtained by changing the delay times D_s of the local in station 1 and 2 from their current values of (3.5, 0.5) to (3, 1) while keeping all other aspects of the solution unchanged. We say that two optimal solutions are *different in the trivial sense* because they differ only in terms of timing (decision variables D_s and Δ) and not in terms of structure (decision variable Y_{st}). On the other hand we say that two optimal solutions are *different in the meaningful sense* if they differ in terms of where passing occurs (decision variable Y_{st}).

We now solve the same problem instance with a modified objective function with weights w_1 =1 and w_2 = 0.001 to show that it has at least two meaningfully different optimal solutions. The weight for the second component of the objective function w_2 is small enough so that it does not interfere with the primary goal of minimizing the cycle length but large enough to be able to identify, among all solutions tied for having the minimum *Interval*, a solution that ties for having the smallest total dwell time for the local train. The resulting optimal timetable is displayed in Table 3.4. Note that the optimal value of *Interval* remains 4 but the total delay for the local train in all stations combined is 0.5+0.5+3.5+3.5=8 min which is much less than before. Also, the passing structure is different than before because passing only occurs in stations 3 and 4. A generic method for finding all meaningfully different optimal solutions for any problem instance is the topic of a future study.

3.4.2 *Illustrative example 2:*

As another numerical example, a railway system with 8 stations is considered. The input data are given in Tables 3.5. Table 3.6 shows an optimal solution for this instance in the form of a timetable. This solution is obtained in about one second. Figure 3.5 shows the same solution displayed in the form of a time-space diagram. As Table 3.6 and Figure 3.5 indicate, the minimum value of *Interval* for this instance is 4 min. Again, this value is higher than the obvious lower bound for *Interval* owing to station headway constraints alone (3.5), and it is also higher than the obvious lower bound for *Interval* owing to main track headway constraints alone (= 2). Note in Table 3.6 that the total delay time for each local train is 17 min which is less (greater) than the maximum (minimum) possible total delay time of 18 (16) minutes. Finally, note that each local train is passed by four express trains–in stations 2, 4, 5, and 8–and each express train passes four local trains–also in stations 2, 4, 5, and 8. Also, the actual number of express trains that come within *hTrack* =1 min of the first local train on any portion of the main track is 4 which less than *maxExpr* = 6 for this example.

Table 3.4 A second optimal timetable for example 1 (passing occurs at stations 3 and 4 only)

Minimum *Interval*		4				
		Station 1	Station 2	Station 3	Station 4	Destination
D_s		0.5	0.5	3.5	3.5	-
	Departure time from origin					
Local	0	1.5-2	5.5-6	11.5-15	18.5-22	23
Express	2.5	4	7.5	13	16.5	17.5
Local	4	5.5-6	9.5-10	15.5-19	22.5-26	27
Express	6.5	8	11.5	17	20.5	21.5
Local	8	9.5-10	13.5-14	19.5-23	26.5-30	31
Express	10.5	12	15.5	21	24.5	25.5
Local	12	13.5-14	17.5-18	23.5-27	30.5-34	35
Express	14.5	16	19.5	25	28.5	29.5
Local	16	17.5-18	21.5-22	27.5-31	34.5-38	39
Express	18.5	20	23.5	29	32.5	33.5

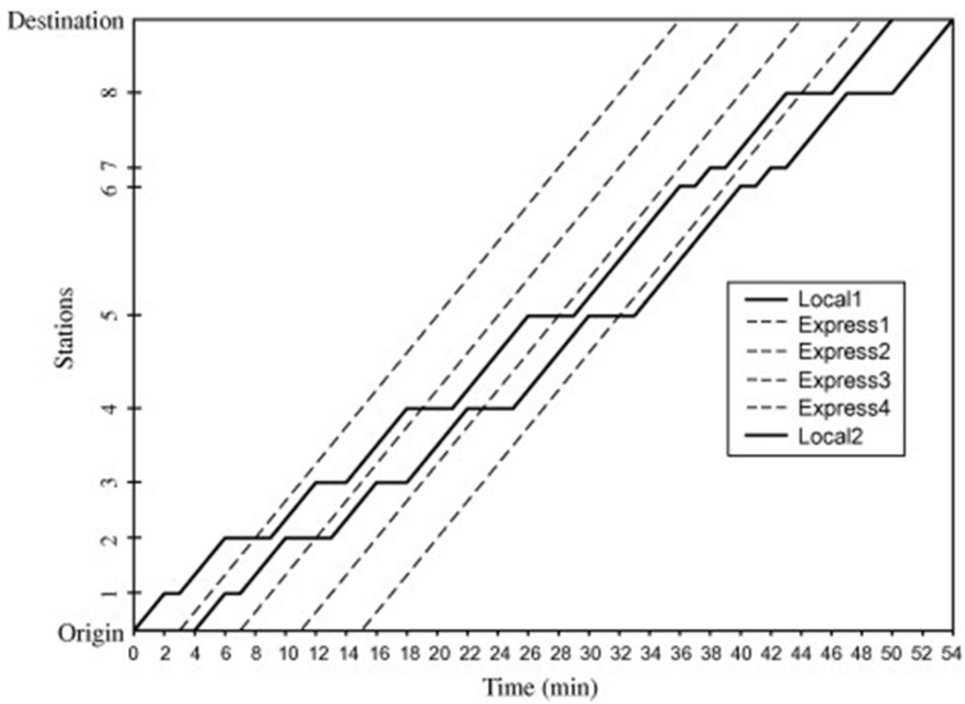


Figure 3.5 Time-space diagram for the optimal timetable in Example 2.

Table 3.5 Input data for example 2

S	*hTrack*	$hStation_s$	*dMax*	w_1	w_2
8	1	0.5 for all *s*	18	1	0

	Station 1	Station 2	Station 3	Station 4	Station 5	Station 6	Station 7	Station 8	Destination	Total
$dMin_s$	1	2	2	3	3	1	1	3	-	16
$trav_s$	2	5	8	12	17	24	25	29	33	-
maxExpr		9								

Table 3.6 An optimal timetable for example 2 (passing occurs in stations 2, 4, 5, 8)

Minimum *Interval*				4						
		Station 1	Station 2	Station 3	Station 4	Station 5	Station 6	Station 7	Station 8	Destination
D_s		1	3	2	3	3	1	1	3	-
	Departure time from origin									
Local	0	2-3	6-9	12-14	18-21	26-29	36-37	38-39	43-46	50
Express	3	5	8	11	15	20	27	28	32	36
Local	4	6-7	10-13	16-18	22-25	30-33	40-41	42-43	47-50	54
Express	7	9	12	15	19	24	31	32	36	40
Local	8	10-11	14-17	20-22	26-29	34-37	44-45	45-46	51-54	58
Express	11	13	16	19	23	28	35	36	40	44
Local	12	14-15	18-21	24-26	30-33	38-41	48-49	50-51	55-58	62
Express	15	17	20	23	27	32	39	40	44	48

3.5. Experimental setup, results, and discussion

In order to study the problem applicability, a number of problems are solved in this section. These problems can be classified as small, medium and large. The size of the model, like other mixed integer programming problems, depends on the number of constraints and decision variables and its difficulty refers to the number of binary decision variables which is a function of number of stations and number of intervals.

We now perform the first numerical study of this problem to demonstrate the effectiveness of the model presented in Section 3.3. The problem instances are defined by the values of the input parameters S, $trav_s$, $dMin_s$, $hTrack$, $hStation_s$, $dMax$, w_k, and M. Table 3.7 shows the parameter values that appear most often in the numerical study. The brackets "$[a, b]$" indicate the set of integers in the closed interval from a to b. Note that $trav_s$ does not appear in the math model and therefore plays only a trivial role in defining each problem instances. Also, $hStation_s = 0.5$, for all stations s, $w_1 = 1$, $w_2 = 0.0001$, and $M = 9999$ in all experiments unless otherwise noted. Thus, the main parameters defining each problem instance are S, $dMin_s$, $hTrack$, and $dMax$. Note that every combination of parameter values satisfying $dMax \geq \sum_{s=1}^{S} dMin_s$ leads to a feasible solution. In the majority of experiments, all input data are rounded to the nearest 0.5, so the optimal value of *Interval* is usually restricted to being a multiple of 0.5. The main objective of the math model is to minimize the cycle time and the secondary, subordinate objective is to minimize the total delay experienced by the local train. These two objectives are pursued in hierarchical fashion by setting w_1=1 and w_2=0.0001 in the majority of experiments. Note that there is no chance for weight w_2 = 0.0001 to interfere with the pursue of the first objective as long as (A) all input data are rounded to the nearest 0.5 and (B) $dMax <$ 5000.

Table 3.8 shows the results from the first set of experiments in which 45 problem instances are solved to optimality. The main parameters defining each instance are given on the left side of the table. In the first 40 (last five) instances, all input data are rounded to the nearest 0.5 (0.01). Note that w_1=1, w_2=0.0001, $hStation_s$=0.5 for all s, and $dMax$ = 1.5 $\times \sum dMin_s + U \sim (1, 10)$ in all instances. In the first 40 instances, this random variable is rounded to the nearest integer. The main aspects of the optimal solution for each instance are given on the right. The results show that very large problems with up to 70 stations can be solved to optimality within reasonable amount of time.

Table 3.7 Most frequently used parameter values in the numerical study

Parameters	Possible Values
S	5, 10, 15, 20, 30, 40, 50,60, 70
$trav_s - trav_{s-1}$	[1,20]
$dMin_s$	[1,5]
$hTrack$	[1, 6]
$hStation$	0.5
$dMax$	[20, 560]
(w_1,w_2)	(1,0.0001)
M	9999

Table 3.9 shows the results from a second set of experiments in which 32 problem instances–organized into eight groups of four instances each– are solved to optimality. The main parameters defining each instance are given on the left side of the table. The main aspects of the optimal solution for each instance are given on the right. The instances range in size from 5 to 70 stations. Each instance with ten or more stations is constructed by adding stations to the end of the corresponding instance with fewer stations. For example, the instance with 50 stations and $hTrack$=2 (in row 21) is

constructed by adding ten stations to the end of the instance with 40 stations and *hTrack*=2 (in row 17); the value of $dMin_s$ for $1 \leq s \leq 40$ are preserved. The results in Table 3.9 shows that, for the particular instances considered, the optimal value of *Interval* is not very sensitive to the problem size. Indeed, the optimal value of *Interval* in most instances with 10-70 stations is equal to the optimal value of *Interval* for the corresponding instance of the next smallest size and with the same *hTrack*. This observation may be a sequence of the particular problem instances considered and may not be a general phenomenon.

Table 3.9 also shows how the CPU runtime required to find an optimal solution is affected by a problem's input parameters. In particular, the integer-group results show that CPU runtime generally increase when *S* and *maxExpr* are simultaneously increased. This is not surprising because the number of decision variables and constraints increases when either *S* or *maxExpr* is increased. Surprisingly, the intra-group results in Table 3.9 show that runtime generally increases (i.e. the problem becomes more difficult) when *hTrack* and *dMax* are simultaneously increased and *S* and *maxExpr* are held constant. This result is somewhat less intuitive because neither the number of decision variables nor the number of constraints increases when *S* and *maxExpr* are held constant. The increase in difficulty corresponds to an increase in the number of options that must be considered in order to verify that a given feasible solution is optimal. We hypothesize that this increase in difficulty reflects an increase in the decision maker's flexibility. If this view is correct, it would imply that the reduction in flexibility created by an increase in *hTrack* is more than cancelled out by the increase in flexibility offered by an increase in *dMax*, assuming *maxExpr* is constant. Overall, it appears that more experiments may be necessary to understand the exact origin of the runtime trends in Table 3.9.

Table 3.8 Results from the first set of experiments ($dMax = 1.5*\sum dMin_s + U(1,10)$).

Problem instance					Optimal solution			
S	$\sum_{s=1}^{S} dMin_s$	*hTrack*	*dMax*	*maxExpr*	*Interval*	No. local trains passed by each express train	Total dwell time for each local train	CPU time (sec)
10	32	1	45	9	5.5	6	33	3
	25	3	41	7	9	4	37	3
	18	4	30	4	13	2	28	3
	24	5	44	5	15	3	44	4
	23	2	39	9	5.5	7	38	3
	24	1	42	8	5.5	5	26	3
	29	2	46	9	6	7	40	5
	24	3	37	7	9	4	36	3
	22	4	42	6	12	3	36	3
	31	5	52	6	17	3	48	3
30	69	4	111	14	16	7	110	10
	73	5	118	12	19	6	117	8
	70	5	107	11	21	5	107	5
	74	3	115	20	11	10	112	12
	68	4	107	14	15	7	105	8
	67	1	103	19	5.5	13	74	5
	77	2	122	23	6.5	19	122	9
	73	1	118	22	5.5	15	81.5	4
	72	2	111	21	7	14	96	12
	78	5	124	13	21	6	123	10
50	116	1	183	34	5.5	23	126	9
	146	4	226	29	16	14	222	358
	137	3	206	35	11	19	204	704
	122	5	185	19	23	8	178	58
	172	4	267	34	14	19	265	222
	134	5	210	21	20	10	206	109
	133	5	209	21	20	10	200	64
	165	1	255	47	5.5	34	185.5	27
	161	5	245	25	22	11	238	214
	151	2	232	43	7	29	200	431
70	194	5	294	30	22	13	291	671
	229	2	345	63	6.5	49	318	1015
	228	1	347	64	6	41	243	157
	209	5	314	32	23	13	304	718
	197	5	299	30	22	13	285	440
	235	2	356	65	7	45	316	2877
	198	3	304	51	11	26	287	14772
	191	4	292	37	17	17	285	1613
	194	4	294	37	17	17	285	2020
	214	5	325	33	22	14	311	1560
31	103.96	1.27	158.27	30	5.59	23	125.31	6
39	112.87	4.34	174.75	21	17.78	9	168.98	24
43	137.55	3.20	214.12	34	12.15	17	207.94	101
56	172.47	2.33	260.43	48	8.20	30	248.96	489
67	196.21	1.89	295.58	54	6.75	44	294.03	2719

Table 3.9 Results from the second set of experiments

Problem instance					Optimal solution			
S	$\sum_{s=1}^{S} dMin_s$	*hTrack*	*dMax*	*maxExpr*	*Interval*	No. local trains passed by each express train	Total dwell time for each local train	CPU time (sec)
5	11	2	12	3	9	1	12	5
		3	18	3	9	2	18	5
		4	24	3	11	2	23	7
		5	30	3	13	2	29	4
10	22	2	24	6	12	2	24	4
		3	36	6	9	4	36	4
		4	48	6	11	4	44	4
		5	60	6	13	4	52	5
15	34	2	36	9	15	2	36	4
		3	54	9	11	4	52	5
		4	72	9	11	6	67	6
		5	90	9	13	6	79	4
30	64	2	72	18	12	6	72	5
		3	108	18	9	12	108	10
		4	144	18	11	12	132	14
		5	180	18	13	13	159	16
40	79	2	96	24	12	7	90	27
		3	144	24	9	15	135	42
		4	192	24	11	15	165	54
		5	240	24	13	16	205	144
50	100	2	120	30	12	9	113	85
		3	180	30	9	20	179	485
		4	240	30	11	19	211	897
		5	300	30	13	20	260	1504
60	121	2	144	36	12	12	144	338
		3	216	36	9	23	207	7755
		4	288	36	11	23	256	6230
		5	360	36	13	23	299	31144
70	140	2	168	42	12	13	164	575
		3	252	42	9	27	244	6332
		4	336	42	11	27	298	254063
		5	420	42	13	28	362	779530

Table 3.10 presents the results of a third set of experiments that aim to isolate, and separately measure, the impact of *hTrack* and *dMax* on the runtime and optimal value. In all instances, S=70, w_1=1, w_2=0, $hStation_s$=0 for all s, and the values of $dMin_s$ for $1 \leq s \leq 40$ are held constant. The results show that runtime generally increases as *dMax* increases with everything else unchanged. However, there are many specific cases that disagree with this general trend. The impact of *hTrack* on runtime is more difficult to distinguish.

Overall, it appears that runtime is concave in *hTrack*. That is, runtime increases as *hTrack* increases for small values of *hTrack* and runtime decreases as *hTrack* increases for large values of *hTrack*. If correct, this observation would indicate that the most difficult problems are characterized by an intermediate value of *hTrack* that is neither very low nor very high.

Figure 3.6 is a graphical representation of the results regarding optimal value in Table 3.10. The impact of *dMax* on optimal value is straightforward. As Figure 3.6 and Table 3.10 demonstrate, the optimal value *Interval* is a decreasing and generally convex function of *dMax* when all other parameters are held constant. That is, the decrease in *Interval* is largest for small *dMax* and is 0 for large values of *dMax* that exceed a certain threshold value that depends on *hTrack*. The very large decrease in *Interval* when *dMax* changes from 140 to 150 can be attributed to the fact that the decision maker has no flexibility when $dMax = 140 = \sum dMin_s$ Indeed, when *dMax*=140, the schedule of the first local train is already fixed $\left(D_s = dMin_s \text{ for all } s\right)$. Thus, the decision maker's only task is to find the starting times of the first express train and second local train such that the cycle length is minimized and the cyclic timetable is feasible. The optimal value of 17 when *hTrack*=1 and *dMax*=140 indicates that between 8 and 9 express trains pass the first local train in the optimal timetable. In other words, the decision maker is lucky to find a feasible timetable that includes a lot of passing. The optimal values of (144, 146, 148, 150, 152) when *hTrack* = (2, 3, 4, 5, 6) and *dMax*=140 indicate that there is no train passing in the optimal timetables for these highly-constrained instances. Figure 3.6 and Table 3.10 also show that *Interval* is an increasing function of *hTrack* when *dMax* and all other parameters are held constant. This result agrees with our intuition that the problem becomes more constrained as the headway on the main line increased.

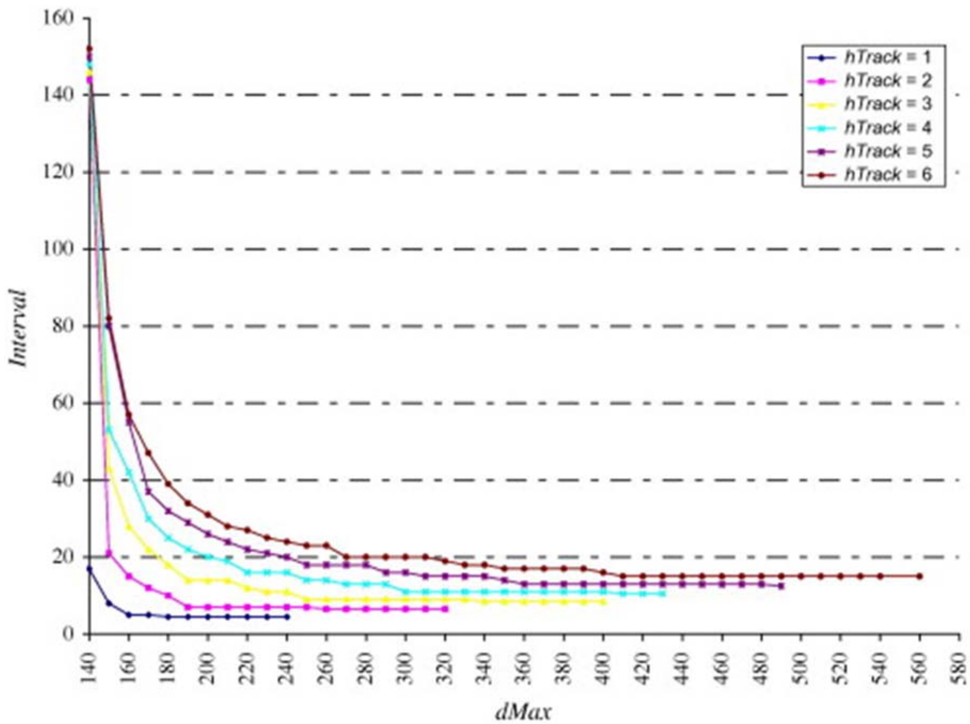


Figure 3.6 Graphical representation of Table 3.10 showing the effect of parameters *dMax* and *hTrack* on decision variable *Interval* for a 70-stations problem instance.

Table 3.10 Effect of parameters *dMax* and *hTrack* on the optimal value (*Interval*) and runtime (in seconds) for a problem instance with 70 stations (w_1 = 1, w_2 = 0, and $\sum dMin_s = 140$)

	hTrack											
dMax	1		2		3		4		5		6	
	Interval	Time	*Interval*	Time	*Interval*	Time	*Interval*	Time	*Interval*	Time	*Interval*	Time
140	17	7	144	3	146	4	148	2	150	2	152	4
150	8	559	21	86	43	114	53	12	80	8	82	10
160	5	348	15	277	28	268	42	44	55	18	57	23
170	5	961	12	708	22	571	30	49	37	25	47	24
180	4.5	942	10	1246	18	1347	25	85	32	42	39	30
190	4.5	1254	7	429	14	798	22	107	29	77	34	41
200	4.5	1043	7	638	14	2357	20	171	26	108	31	48
210	4.5	898	7	1886	14	4471	19	335	24	118	28	54
220	4.5	2247	7	6140	12	1756	16	381	22	186	27	111
230	4.5	1408	7	3454	11	3600	16	1950	21	240	25	227
240	4.5	2006	7	9782	11	17972	16	2939	20	336	24	231
250			7	6692	9	8797	14	1043	18	420	23	590
260			6.5	3302	9	1548	14	2199	18	1780	23	1104
270			6.5	1669	9	3357	13	2492	18	1465	20	246
280			6.5	4094	9	4248	13	5433	18	2637	20	387
290			6.5	14114	9	3794	13	10870	16	3377	20	522
300			6.5	24915	9	1654	11	2381	16	4310	20	1115
310			6.5	10874	9	1948	11	1515	15	4017	20	3077
320			6.5	91560	9	2535	11	1435	15	10968	19	2614
330					9	2430	11	7734	15	4288	18	1507
340					8.5	7307	11	2036	15	27004	18	10282
350					8.5	7835	11	1602	14	13897	17	5249
360					8.5	17854	11	1866	13	6351	17	8737
370					8.5	5858	11	2292	13	1085	17	21388
380					8.5	4645	11	13172	13	2396	17	10065
390					8.5	16058	11	5910	13	5713	17	18195
400					8.5	5691	11	10404	13	3415	16	15091
410							10.5	3509	13	2537	15	5555
420							10.5	6609	13	14177	15	2758
430							10.5	28285	13	26895	15	5248
440									13	4377	15	3463
450									13	3269	15	1532
460									13	5213	15	3872
470									13	2505	15	12885
480									13	61151	15	31523
490									12.5	46230	15	3125
500											15	22169
510											15	6808
520											15	2521
530											15	19522
540											15	2079
560											15	60948

Table 3.11 shows the results from a fourth set of experiments in which 27 problem instances–organized into three groups of nine instances each– are solved to optimality. The main parameters defining each instance are given on the left side of the table and the main aspects of the optimal solution for each instance are given on the right. These experiments investigate the effect of the number of stations on the optimal value when *dMax*, *hTrack*, and $\sum dMin_s$ are held constant. In other words, we explore the effect of having (A) few stations with long dwell times at each station versus (B) many stations with short dwell times at each station on the optimal value of *Interval*. Put another way, we consider how the number of stations, over which the same total minimum dwell time is $\sum dMin_s$ spread, affect the optimal value of *Interval*.

The intra-group results in Table 3.11 reveal an interesting phenomenon, namely that the optimal value of *Interval* is generally convex in the number of stations over which same $\sum dMin_s$ value is spread. In particular the optimal value of *Interval* is smallest when the number of stations over which the same $\sum dMin_s$ value is spread neither very small nor very large, but rather an intermediate value. Not surprisingly, the values in the "number local trains passed by each express train" column have the opposite trend as those in the *Interval* column. Indeed, the smallest optimal value of *Interval* is attained precisely when the number of local trains passed by each express train in the optimal solution reaches its highest value. Regarding CPU runtime the intra-group results in Table 3.11 show that the runtime generally increases–i.e. the problem becomes more difficult–as the number of stations *S* increases. Also, the inter-group results show that the problem generally becomes more difficult when *dMax* and $dMin_s$ are increased in the same proportion and the other parameters are unchanged.

Table 3.11 Effect of number of stations on the optimal value when *dMax*, *hTrack* and $\sum dMin_s$ are held constant

Problem Instance				Optimal Solution				
dMax	*hTrack*	$\sum_{s=1}^{S} dMin_s$	*S*	*Interval*	Δ	No. local trains passed by each express train	Total dwell time for each local train	CPU time (sec)
100	3	70	5	14.5	3	5	70	8
			10	13	4	7	85	7
			15	10	3	10	100	8
			20	9	6	10	90	9
			30	10	7	10	100	12
			40	14	9	7	98	24
			50	15	12	6	97	191
			60	15	12	6	98	181
			70	16	12	6	100	371
200	3	140	5	28.5	3	5	140	8
			10	14.5	3	10	140	7
			15	13.5	3	15	195	9
			20	12	3	15	176	9
			30	10	3	20	196	27
			40	10	5	20	198	63
			50	8	3	25	200	84
			60	14	9	13	186	1313
			70	14	5	14	196	3958
300	3	210	5	42.5	3	5	210	9
			10	21.5	3	10	210	7
			15	15.5	3	15	223	8
			20	15	3	15	222	9
			30	12	3	20	237	12
			40	12	6	20	237	166
			50	9	3	25	225	46
			60	9	4	32	286	29107
			70	12	6	23	279	26784

Table 3.12 shows the results from a fifth set of experiments that show the effect of the sequence of the same $dMin_s$ values on the optimal value of *Interval* assuming all other parameters are unchanged. These experiments consider 48 problem instances. In every instance, S=15, w_1=1, w_2=0.0001, $hStation_s$=0.5 for all s, and the same set of $dMin_s$ values are used. The exact sequence of $dMin_s$ values considered for each instance is given in the left side of the table and the optimal value of *Interval* under four different combinations of values for *dMax* and *hTrack* are displayed on the right. Interestingly, the sequence of $dMin_s$ values affects the optimal value of *Interval* in some cases but not in

others. When (*dMax*, *hTrack*) = (84, 3) or (104, 4), the optimal value of *Interval* is insensitive to the order of the $dMin_s$ values. However, when (*dMax*, *hTrack*) = (84, 2) or (104, 3), the optimal value of *Interval* is sensitive to the order of the $dMin_s$ values. Based on these results, we hypothesize that the optimal value of *Interval* becomes more sensitive to the sequencing of the $dMin_s$ values as *hTrack* decreases or *dMax* increases. The results seem reasonable because the size of feasible region increases–i.e. the problem becomes less constrained–as *hTrack* decreases or *dMax* increases. More experiments are probably needed to understand the exact mechanism by which the sequence of the $dMin_s$ values impacts the optimal values of *Interval*.

Table 3.12 Effect of the sequence of $dMin_s$ values on *Interval* with all other parameters unchanged.

Optimal value of *Interval* is shown in the shaded region															*dMax* 84		*dMax* 104	
$dMin_s$ for s =															*hTrack*		*hTrack*	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	2	3	3	4
1	1	2	2	2	4	4	4	4	4	4	4	4	5	5	8	10	10	12
1	1	5	5	4	4	4	4	4	4	4	4	2	2	2	8	10	10	12
2	2	2	1	1	4	4	4	4	4	4	4	4	5	5	8	10	10	12
2	2	2	5	5	4	4	4	4	4	4	4	4	1	1	8	10	10	12
5	5	1	1	2	2	2	4	4	4	4	4	4	4	4	8	10	10	12
5	5	2	2	2	4	4	4	4	4	4	4	4	1	1	8	10	10	12
5	2	1	1	4	4	4	4	4	5	2	2	4	4	4	7	10	10	12
4	4	4	4	4	5	2	1	1	5	2	2	4	4	4	7	10	10	12
2	1	5	4	4	4	4	1	4	5	4	4	4	2	2	6.5	10	8.5	12
4	4	4	4	4	2	1	1	5	5	2	2	4	4	4	6.5	10	8.5	12
5	2	1	4	4	4	4	1	4	5	2	2	4	4	4	6	10	8	12
5	2	1	4	4	4	4	1	4	5	4	2	2	4	4	6	10	8	12

Figure 3.7 shows the results from a final set of experiments that explore the trade-off between the minimization of *Interval* and the minimization of the total dwell time of the local train at all stations combined. This trade-off can be analyzed by adjusting the objective function weights w_1 and w_2. Figure 3.7 displays Pareto-optimal solutions for problem instances with 10-20 stations for six different values of *hTrack*. The horizontal

(vertical) axis indicates the local train's total dwell time in minutes (the value of *Interval*) in the optimal solution. Graphs A-F pertain to a 10-station problem when *hTrack* equals to 1, 2, 3, 4, 5, and 6 respectively. Graphs G-L (M-R) show the results for a 15- (20-) station problem with the same *hTrack* values respectively. The steepness of the left portion of each curve indicates that in many cases it is possible to obtain substantial reduction with respect to objective 1 (the value of *Interval*) with only a small increase in objective 2 (total dwell time of the local train at all stations combined). The converse is also true. Indeed, in many cases it is possible to obtain substantial reduction with respect to objective 2 at the cost of little or no increase in objective 1.

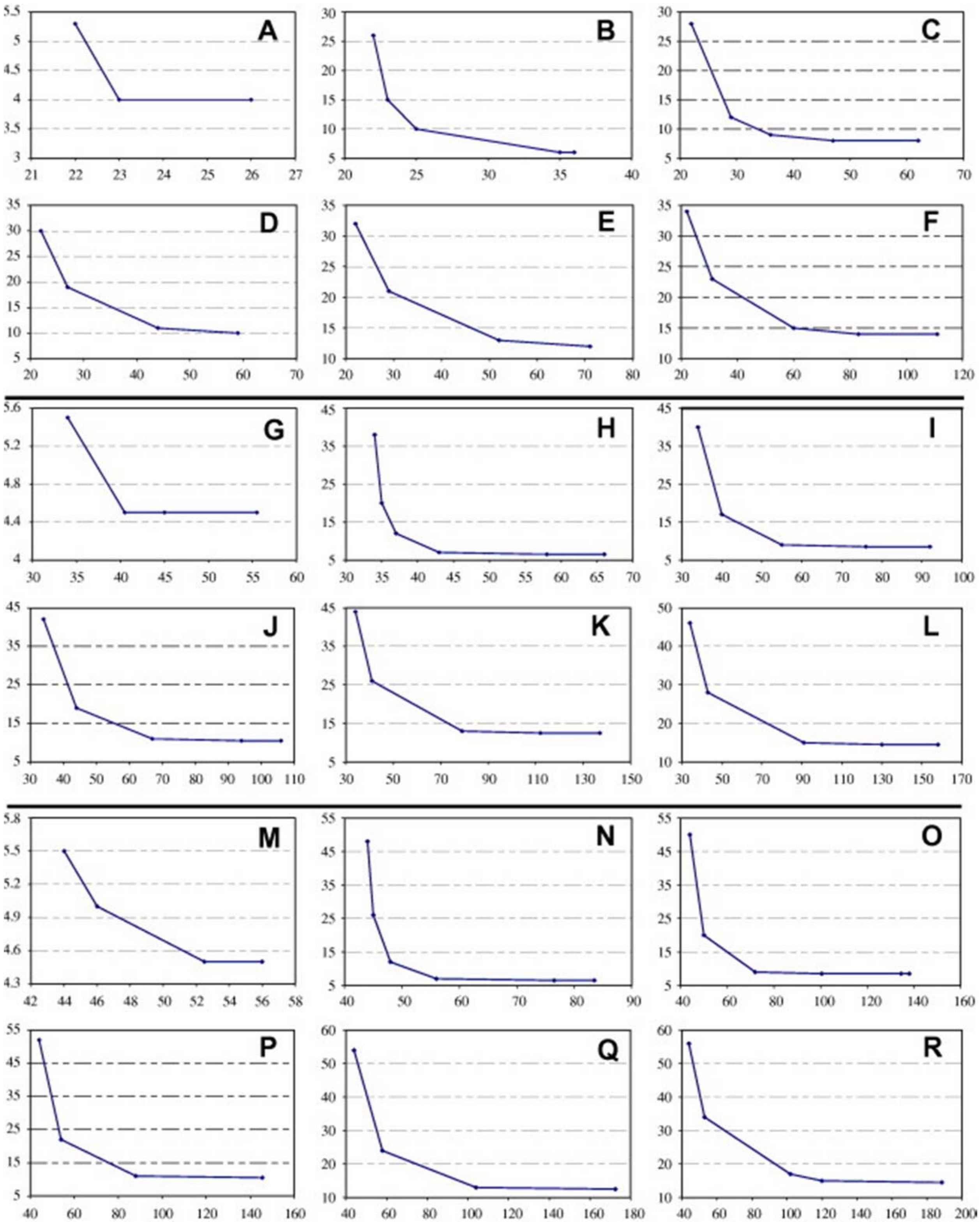


Figure 3.7 Pareto optimal solutions for problem instances with 10-20 stations for six different *hTrack* values (horizontal axis = local train's total dwell time in minutes, vertical axis = *Interval*). Graphs A-F relate to a 10-station problem when *hTrack* equals to 1, 2, 3, 4, 5, and 6 respectively. Graphs G-L relate to a 15-station problem and graphs M-R relate to a 20-station problem with the same *hTrack* values, respectively.

3.6. Alternate optimal solutions

As mentioned earlier in this chapter, one characteristic of this timetabling problem is the case of multiple optimal solutions (alternate optima) with respect to the first and main objective function of the problem, i.e. *Interval*. Also as discussed, by multiple optimal solutions we mean that two optimal solutions are *different in the meaningful sense*, i.e. passing occurs at different station or equivalently the locals meet different number of express trains in its journey from origin to destination. In other words, in two sets of solutions if the numbers of expresses are equal, express trains must pass the local at different stations but the two solution sets have the same optimal value of *Interval*; or the local meets different number of expresses in two different set for the same value of *Interval*.

Now, the question is how to find other optimal solutions. This is of great importance from planning and decision making point of view. Since our emphasis is on the main and primary objective, from this point toward the end of this chapter we just focus on the single objective problem, rather than a multiple objective. But, we will make use of the secondary objective to find other optimal solutions. As defined, in our model with respect to the first objective, those solutions are considered as alternate which have different set of binary variables. Therefore, one method of finding alternate optima is equivalent to the weighted sum approach for multiple objective functions where the alternate optima can be obtained by varying weights (Cohen, 2003).

Consider the following maximization problem and suppose that $w_k > 0, \forall k = 1,\ldots,h$, and w_k=0 for $\forall k = h+1, h+2,\ldots, p$.

Maximize

$$Z\left(x_1,x_2,\ldots,x_n;w_1,\ldots,w_p\right)=\sum_{k=1}^{p}w_kZ_k\left(x_1,x_2,\ldots,x_n\right) \tag{3-14}$$

Subject to

$$\left(x_1,x_2,\ldots,x_n\right)\in F_D \tag{3-15}$$

where (3-15) means that the constraints of the model must satisfy F_D or solution space of the model. Suppose that we solve the above model and find alternate optimal solutions that give $Z_k\left(x_1,x_2,\ldots,x_n\right)=Z_k^*$ for $k=1,\ldots,h$. Then the following problem can be solved to find alternative optima.

Maximize

$$\sum_{k=h+1}^{p}w_kZ_k\left(x_1,x_2,\ldots,x_n\right) \tag{3-16}$$

Subject to

$$\left(x_1,x_2,\ldots,x_n\right)\in F_D \tag{3-17}$$

$$Z_k\left(x_1,x_2,\ldots,x_n\right)=Z_k^* \qquad k=1,\ldots,h \tag{3-18}$$

The values of w_k are chosen in various combinations to find an approximation of other alternate optima. For our timetabling problem, a similar method is used.

According to this method, for our mixed integer program three groups of instances with various numbers of stations are considered (10, 15 and 20 stations). For each group six different values of *hTrack* are considered. In order to find other optimal solutions, the second expression, total dwell times at stations, is added to the main objective, *Interval*.

The results are given in Table 3.13. For each 18 instances in Table 3.13, eight problems are solved with different coefficient for total dwell times, while keeping *Interval* coefficient fixed at 1. The coefficient set used are: (1, 0), (1, 0.0001), (1, -0.0001), (1, 0.001), (1, -0.001), (1, 0.01), (1, -0.01) and (1, 0.1). Among these 8 cases for each instances, only for 10-station group the coefficient (1, 0.1) had the same objective value for *Interval* as other 7 sets. In other words, for 10-stations group we considered 48 instances, 8 for each level of *hTrack*, and for two other groups 42 instances were considered.

After showing the existence of alternate optima and describing a method based on weighted sum approach, we will present a procedure for the equivalent single objective problem (i.e. w_1 =1 and w_2 =0) of our mixed integer linear program for cyclic train timetabling. But before doing so, let us look at some previous works done on finding alternate optimal solutions for mixed integer programming models. There are some works in the literature devoted to development of techniques of generating other (near) optimal solutions. Among them we can name the work by Lee et al. (2000) which generates multiple optimal solutions for LP, Glover et al. (2000) use an MIP-based heuristic to generate multiple optimal solutions for MIP, and Danna et al. (2007) develops three heuristics namely one-three algorithm, a branch-and-bound-based algorithm, an algorithm based on MIP heuristics and an algorithm generalized from previous approach which generates multiple solutions sequentially. These three approaches are used in CPLEX. None of these approaches are applicable to our problem due mainly to the structure of the model and binary decision variables and their relationship. Therefore, a similar method based on fixing variables, called "fix-and-cut", is developed to find other optimal solutions which will be discussed hereafter.

Like other algorithms for generating optimal solutions, in our method, we first solve the problem to optimality in order to find the optimal value of the primary objective, *Interval*. Then, like weighted sum approach, the optimal value of the objective function is added to the model as a constraint. In the next step a binary decision variable (Y_{st}) is forced to take on value 1. This is very close to the well-known method proposed by Balas and Jeroslow (1972) for finding other optimal solution for especial case of binary variable by adding one constraint to cut off the current solution and make it infeasible. There are differences between the proposed method and the one proposed by Balas and Jeroslow (Balas and Jeroslow, 1972) though. These differences are because of the special structure of the train timetabling mathematical model which will be discussed shortly.

Based on this brief introduction, we are now ready to present the technique for finding all optimal solutions. This technique works in three steps:

Step 1:

Solve the problem and find the optimal solutions to binary variables and optimal value of objective function (Interval)

Step 2:

Add the objective function to the constraints and take binary decision variables into the basic by changing their values from 0 to 1. If this change leads us to a feasible, thus optimal, solution and it is new add it to Multiple Optimal Solutions;

if it becomes infeasible delete that decision variable from further consideration, thus fathomed.

Step 3:

Continue until there is no binary decision variable to be added to the basic.

For illustration purposes and without loss of generality, consider an instance with 5 intermediate stations with the following binary solutions shown

$Y_{11} = 1$

$Y_{21} = 1$

$Y_{31} = 1$, $Y_{32} = 1$

$Y_{41} = 1$, $Y_{42} = 1$, $Y_{43} = 1$

$Y_{51} = 1$, $Y_{52} = 1$, $Y_{53} = 1$, $Y_{54} = 1$

This output says that 4 expresses pass the first local at stations 1, 3, 4 and 5. As can be seen from the solution $Y_{s1} = 1$, $\forall s = 1,\ldots,5$. It is true since once an express train passes the local it cannot be overtaken by the same local, meaning that once an express arrives at the merge point of station *s*, it will arrive at all stations after station *s*. So all arrays should equal 1 in the column related to the express train after row s. Therefore, in the above solution, since the first express passes local at station 1, all entries in the first column should be 1. The second express passes local at station 3, so in the second column all entries from row three should be 1 inclusive. Similarly, the third and fourth expresses pass the local at station 4 and 5 respectively. Therefore, in the third column all entries after row 4, and in the fourth column the array in row 5 should be 1 (Note that the secondary parameter *maxExpr* is 4 for this typical example, therefore the size of matrix is 5×4).

From above description it can be revealed that binary decision variables have a special characteristics enforcing that a binary decision variable, say Y_{st} can take on 1 only if $Y_{s-1,t-1}$ has taken value 1. This is because at each station at most one express can pass local while it dwells there, which is based on the assumption that in each interval there is only one express train. For instance, in the above solution the first express pass the first local at station 1 (Y_{11} =1) and the second express passes it at station 3 ($Y_{32} = 1$), but since $Y_{11} = 1$, it possible to consider another case where ($Y_{22} = 1$). Therefore, we fix Y_{22} at 1 and add Y_{22}=1 and objective function to the constraints, if it returns another solution, which is optimal; we consider the just generated solution as a new and different optimal solution with regard to Y_{st}. Solving this new problem gives us the following optimal solution and since it is different from the above we can consider it as a new optimal solution.

$Y_{11} = 1$

$Y_{21} = 1$, $Y_{22} = 1$

$Y_{31} = 1$, $Y_{32} = 1$

$Y_{41} = 1$, $Y_{42} = 1$, $Y_{43} = 1$

$Y_{51} = 1$, $Y_{52} = 1$, $Y_{53} = 1$, $Y_{54} = 1$

In the new optimal solution four expresses pass the first local at stations 1, 2, 4 and 5.

To restate the procedure used in the example, suppose that we can show the optimal binary decision variables in the matrix form as

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Considering the initial optimal solution as the root of branching tree, we start from the upper left corner and take in (out) each binary decision variable to (of) basis. If it results in a new solution, we consider it as a new optimal solution and branch out from that node down the tree; otherwise we do not need to consider any node downward, so fathomed. This process is closely similar to branch and bound schematically. This technique was applied to the first group of problems in Table 3.13 and results are given in the Appendix (Table A.1 and Figure A.1).

3.7. Discussion

This chapter has extended the work of Bergmann (1975) to investigate the capacity of a single track, unidirectional rail line that adheres to a cyclic timetable. Within the railway operations literature, the work by Bergmann (1975) appears to be unique in that is the only article (1) consider cyclic timetabling problem in which the minimization of cycle length is the primary objective; (2) present a linear formulation of a cyclic timetabling problem in which the length of the cycle is a decision variable; and (3) focus on the alternate definition of railway capacity in which capacity is the minimum cycle length that can be feasibly accommodate a given number of trains over a given section of track in each cycle. The purpose of this chapter was to restate the problem from Bergmann

(1975) using improved notations, to modify the mathematical model by adding a second objective and removing unnecessary variables, and to perform the first numerical analysis of this problem by considering hundreds of randomly generated instances with up to 70 stations.

The experimental results can be summarized as follows. First, our ability to solve every problem instance to optimality in a reasonable amount of time using IBM ILOG CPLEX demonstrates the effectiveness of the model. Second, the problem becomes more difficult when (A) either of the parameter *S* or *maxExpr* are increased and everything else in unchanged; (B) parameter *hTrack* and *dMax* are simultaneously increased in such a way that neither the number of decision variables or the number of constraints (which are functions of *S* and *maxExpr* only) increases; or (C) *dMax* increases and everything else is unchanged. The case (A) is expected to happen, since increasing either *S* or *maxExpr* (or both) will increase the decision point or mathematically increase the size of the problem mathematically. Cases (B) and (C) are the direct results of expanding the solution space which requires more computation time. Third, problem difficulty appears to increase as the parameter *hTrack* increases for small values of *hTrack* but appears to decrease as *hTrack* increases for large values of *hTrack*. Fourth, the optimal value of *Interval* is smallest when the number of stations over which the same $\sum dMin_s$ value is spread is neither very small nor very large, but rather an intermediate value. Fifth, the sequence of $dMin_s$ values affects the optimal value of *Interval* in some cases but not in others. In particular, the optimal value of *Interval* seems to become more sensitive to the sequencing of the $dMin_s$ values as *hTrack* decreases or *dMax* increases. Sixth, Pareto-optimal solutions that explore the trade-off between cycle length and the local train's dwell time at all stations combined can be constructed by adjusting the objective function weights w_1 and w_2. Finally, since the problem has alternate optima, two methods are explored and described in order to find other optimal solutions. The first method is based on the weighted sum approach while the second one is based on the so called “fixed-and-cut” procedure.

Table 3.13 Number of optimal solutions for selected instances

	S																	
	10						15						20					
hTrack	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
dMax	72	92	112	142	172	212	104	134	154	194	244	294	144	174	234	284	334	374
No. Optimal Solutions Found	3	4	4	5	3	3	3	3	3	3	2	4	3	5	4	5	4	4

Chapter 4

A mixed integer linear program for uni-directional cyclic train timetabling and platforming with homogenous rolling stock

4.1. Introduction

In chapter 3, we formulated a single-unidirectional track railway system with one siding at each station where local trains can stop to let passengers board and alight. Also, we considered only two types of trains: express and local. The express train will not make a stop once it departs from the origin until it arrives at final station, known as destination. However, local train must stop at every station between origin and destination.

In this chapter we generalize the problem to investigate the capacity of a single track, unidirectional rail line that adheres to a cyclic timetable. A set of intermediate stations lies between an origin and destination with one or more parallel sidings at each station. A total of T train types—each with a given starting and finishing point and a set of known intermediate station stops—are dispatched from their respective starting points in cyclic fashion, with one train of each type dispatched per cycle. Two mixed integer linear programs (MILP) are developed in order to schedule the train arrivals and departures at the stations and assign train types to tracks (platforms) in the stations so as to minimize the length of the dispatching cycle and minimize the total stopping (dwell) time of all train types at all stations combined. Sets of constraints considered include: a) minimum dwell time for each train type in each of the stations in which it stops, b) a maximum total dwell time for each train type, and c) headway considerations on the main line and on the tracks in the stations. The applicability of the model is shown by solving hundreds of randomly generated and real-world problem instances [with 4-33 stations and 2-11 train types] to optimality in a reasonable amount of time using IBM ILOG CPLEX 11.2 and 12.4.

As mentioned earlier, the current study generalizes the problem in chapter 3. This generalization is carried out through three different ways. First, the current study considers any number of train types per cycle. Second, we allow stations to have more than one siding. Third, we allow trains to start or end at intermediate stations. As we shall demonstrate, the generalization regarding the first aspect leads to a significantly more complex mathematical model than what is presented in chapter 3. In addition, the generalization regarding the second aspect introduces a train platforming component that is modeled and investigated for the first time in this dissertation. That is, in chapter 3 we consider a pure train timetabling problem, while the current study considers a combined train timetabling and platforming problem.

4.2. Brief literature review

As discussed in Chapter 2 cyclic train timetabling problem has been addressed by less than 50 articles in the literature and as pointed out by Harrod (2012) most of them rely on the Periodic Event Scheduling Problem (PESP), where all calculation are carried out modulo T, the cycle time. The minimization of the cycle length requires the cycle time T to be a decision variable. However, within the PESP paradigm, this leads to a quadratically constrained formulation. By avoiding the PESP framework, the current study manages to create a linear formulation for a periodic timetabling problem in which the cycle time is a decision variable and also an objective to be minimized. Surprisingly, to our knowledge, Bergmann (1975) and Heydar et al. (2013) are the only two studies to consider a cyclic train timetabling problem in which the minimization of the cycle length is the primary objective.

This chapter also relates to train platforming problem in that trains need to be assigned to tracks when they stop in stations that have more than one siding and/or platform. Train platforming (i.e. track allocation, train pathing) refers to the allocation of track to trains over time (Lusby et al. 2011a). This allocation may occur within stations (i.e. train platforming) or along portions of track between stations. This chapter considers the simplest form of the train platforming problem in which we decide the platform (track, siding) visited by a train in a station under the assumption that each platform specifies a unique predetermined path through the station over time. Recent contributions that consider train routing or platforming but not timetabling include Zwaneveld et al. (1996), Zwaneveld et al. (2001), Carey and Carville (2003), Cornelsen and Di Stefano (2007), Billionnet (2003), Chakroborty and Vikram (2008), Chung et al. (2009), Lusby et al. (2011b), Caprara et al. (2011), and Demange et al. (2012).

Recent papers considering non-cyclic, combined timetabling and routing/platforming problems include Carey (1994), Ghoseiri et al. (2004), Carey and Crawford (2007), Lee and Chen (2009), and Yang et el. (2010). Other integration includes timetable planning and rolling stock in rapid transit networks (Cadarso & Marín, 2012).

Almost no studies consider a cyclic, combined train timetabling and routing/platforming problem. Indeed, according to Liebchen and Möhring (2008), “routing of trains through stations or even alternative tracks” is “definitely beyond the scope of the PESP.” To our knowledge, Shafia et al. (2012) is the only study besides the current study to consider a cyclic, combined train timetabling and platforming problem. That study differs from the current one in at least five ways. First, it adopts a fuzzy logic methodology. Second, it assumes a fixed-block signaling system which allows the authors to apply job shop scheduling models to the problem of interest. Third, train routing is implicitly performed using station capacity constraints (constraints 20-23) that limit the number of trains that can simultaneously dwell in a station. Fourth, a heuristic method is needed to consider large problems with 15 or more intermediate stations. Fifth, some stations (e.g. Kashan and Forudgah in Fig. 15) are not served by any trains. The current study, on the other hand, introduces an MILP model; does not assume a fixed-block signaling system; explicitly assigns trains to tracks in each station; is able to obtain optimal solutions by direct application of the MILP model to large instances with more than 30 intermediate stations; and assumes that all stations are served by at least one train type.

Our inspection of Shafia et al. (2012) revealed several problems. For example, decision variables z_{ijk} and z'_{ij} are undefined. Also, index i is used in two ways in constraint 22—to indicate the number of trains that simultaneously dwell in a station and to indicate a particular train. Variable δ_{ijk}, which is defined as a binary variable that equals 1 if trains i and j dwell in station k during overlapping time intervals, does not appear in any constraints. Rather, the variables δ_{ij}, δ^{k}_{ijA}, and δ^{k}_{ij}, which are not defined, appear within constraints 20-22. Despite this confusion, it appears that the constraints where δ appears (constraints 20-22) are incorrect. Here, it is possible for all δ to be zero regardless of the values of the other decision variables. In other words, even if other variables indicate that train overlapping occurs in one or more stations, all δ variables may take the value of zero.

4.3. Problem definition

We now formally introduce the problem, and then two mixed integer linear programs will be presented. Figure 4.1 shows the topology of the railway system investigated in this chapter. We consider a single track, unidirectional rail line that consists of an origin, destination, S intermediate stations lying between the origin and destination, and a set of parallel sidings in each station that accommodate trains stopping in that station. A total of P platforms (sidings, tracks) exist in the entire rail line ($P \geq S$) and there is at least one siding in each station. The platforms in each station are arranged in parallel and are capable of accommodating all train types that reside there. $P = S$ means that every station has exactly one siding and $P > S$ means that at least one station has two or more sidings. Let s_p denote the station in which platform p resides ($1 \leq p \leq P$) and let p_s denote the set of platforms belonging to station s ($1 \leq s \leq S$).

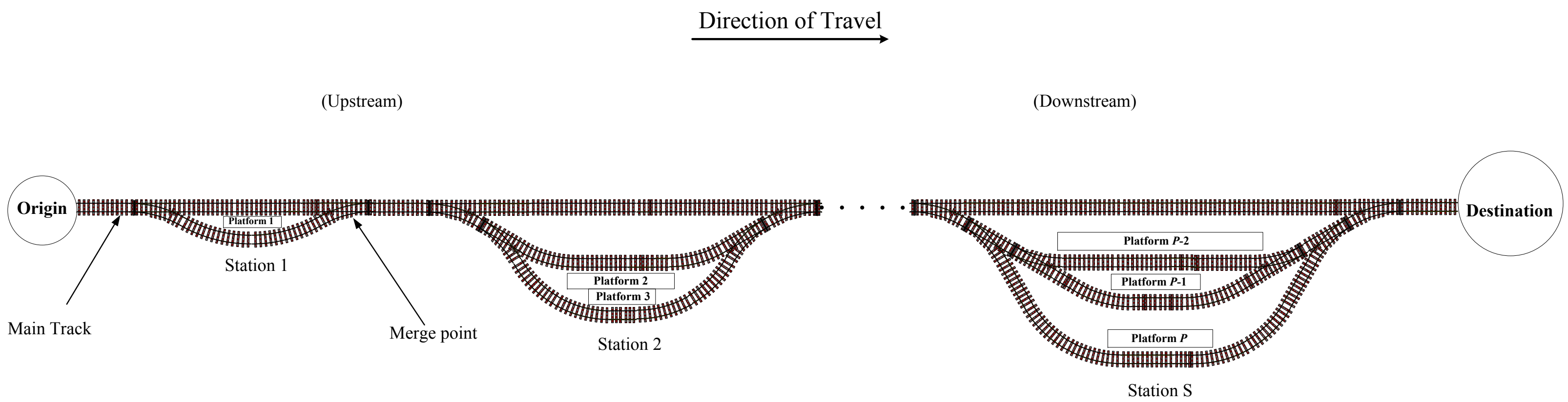


Figure 4.1 Topology of the railway system investigated in this chapter

A total of T train types—each with a given starting and finishing point and a set of known intermediate station stops—are dispatched from their respective starting points in cyclic fashion, with one train of each type dispatched per cycle. The starting point (origin station, $oStation_t$) of train type t may be the origin (i.e. "station 0") or any intermediate station. The finishing point (destination station, $dStation_t$) of train type t may be any intermediate station after $oStation_t$ or the destination (i.e. "station S+1"). The given binary parameter w_{ts} defines each train type line plan and denotes whether or not train type t stops in station s. That is, w_{ts} = 1 (0) means that train type t stops (does not stop) in station s. The train type may relate to stopping frequency, train-station compatibility, or some other differentiating characteristic between train sets. For example, there might be five train types dispatched per cycle: a super express train that makes no stops between the origin and destination; a limited express train that makes two stops between the origin and destination; an express train traveling from the origin to the destination that stops at every other station; a local train with ($dStation_t \neq$ the destination) that stops at all stations between the origin and the point halfway along the rail line; and a local train with ($oStation_t \neq$ the origin) that stops at all stations between the point halfway along the rail line and the destination.

The single track makes it impossible for one train to pass another if both trains are on the main track. However, there is at least one siding in each station, so it is possible for a train on the main line to pass a train that is stopped at a station. Regarding passing, we assume that a through train's passage of a station is entirely unobstructed by any trains that are stopped at the station. Further, each station siding is sufficiently long so that deceleration and acceleration by a train moving into or out of a station does not interfere with the operations of the through trains that do not stop at the station. Therefore, these required times are included in the minimum dwell times, $dMin_{ts}$.

Each of the T train types is dispatched from its starting point once per cycle and the cycle length, a decision variable, is *Interval*. Let D_{ts} denote the departure time of the original (i.e. first) train of type t from station s ($oStation_t \leq s < dStation_t$) and A_{ts} denote the arrival time of the original train of type t at station s ($oStation_t < s \leq dStation_t$). The cyclic nature of the timetable means that the departure (arrival) time of the ith train of type t from (at) station s is given by $[(i\text{-}1) \times Interval + D_{ts}]$ ($[(i\text{-}1) \times Interval + A_{ts}]$). The departure time from station s is the time when a train reaches the merge point—the point at which the sidings in station s return to the main track—just after station s. Note that $D_{t,oStation_t}$ denotes the departure time of the original train of type t from its starting point. Without loss of generality, we assume that $0 \leq D_{t,oStation_t} \leq Interval$ for all t. This constraint can be inferred from the assumption that there is one train dispatched in each cycle. The cyclic nature of the timetable means that all trains of type t are identical except for their departure times from their starting point which is one cycle length apart from the original train type t.

All trains travel at the same speed while on the main track. Let $trav_s$ be the traveling time for all train types along the main line between stations s and s+1. Thus $A_{t,s+1} = D_{ts} + trav_s$ for all train types t that travel between stations s and s+1. Denote by $dMin_{ts}$ the minimum required delay or increase in travel time for trains of type t due to stopping at station s rather than passing it. This is the minimum amount of time that the train type must spend decelerating, unloading passengers, loading passengers, and accelerating on the siding beside station s in order to serve station s. For simplicity, we refer to this term as the minimum stopping (delay, dwell) time for train type t at station s. Note that (D_{ts} - A_{ts}) is the actual stopping (delay, dwell) time of the original train (and therefore all trains) of type t at station s. D_{ts} and A_{ts} are decision variables whose values must satisfy (D_{ts} - A_{ts}) $\geq dMin_{ts}$.

The primary objective is to minimize *Interval* subject to four types of constraints. First, the actual dwell time (D_{ts} - A_{ts}) of a train in a station must be greater than or equal to the minimum required dwell time $dMin_{ts}$. Second, the total dwell time of train type t at all stations combined, $\sum_{s=oStation_t+1}^{dStation_t-1}(D_{ts} - A_{ts})$, may not exceed $dMax_t$. Third, trains on the portion of the main track between stations s and s+1 must be separated by a minimum headway of $hTrack_s$ minutes ($0 \leq s \leq S$). Finally, the departure time of a train from platform (track) p and arrival time of the next train after it into platform p must be separated by a minimum headway of $hPlatform_p$ minutes. For simplicity, we assume that a train's "departure time" from a station occurs at the very end of its dwell time in the station, and we assume that a train's "arrival time" into a station occurs at the very beginning of its dwell time in the station. The secondary objective is to minimize the total delay $\sum_{t=1}^{T}\sum_{s=oStation_t+1}^{dStation_t-1}(D_{ts} - A_{ts})$ experienced by all train types at all stations combined.

As defined and discussed in Chapter 3, the primary objective directly relates to track capacity. Indeed, in order to determine the maximum capacity of a single track, we can either determine the maximum number of trains that can be dispatched within a period (or "cycle") of given length, or we can determine the minimum cycle length such that a given number T of trains can feasibly be dispatched within one cycle. The first objective deals with the capacity of entire line whereas the second objective only considers each train type characteristics individually.

Figure 4.2 presents a time-space diagram of cyclic train timetable described above in which $T = 3$. Each train type is depicted using a different style line (e.g. solid, dashed). The slopes of all lines between stations are equal because all trains move at the same speed on the main line. Note that most of the terms introduced in the preceding paragraphs—including *Interval,* the length of the dispatching cycle—appear in the figure. We now make a few additional comments regarding the problem. First, we assume there is no limit on the number of trains available. That is, rolling stock components (i.e. locomotives, railcars, train sets) are assumed to be available whenever and wherever they are needed. Second, we assume the origin and destination have unlimited capacity to accommodate trains. In other words, we ignore any constraints on the operations at and before the origin, and at and after the destination. However, the headway constraints on the main track immediately after the origin and immediately before the destination are considered. Likewise, for any train type t that makes a "short" journey (such that ($oStation_t \neq$ the origin) or ($dStation_t \neq$ the destination)), we assume that the train type does not use up capacity at or prior to $oStation_t$, or at or after $dStation_t$. Thus, we only consider the train type's impact on the capacity on the main line between, and the stations that are strictly between, $oStation_t$ and $dStation_t$. Here again, the origin and destination have been ambiguously defined so either or both of these locations may represent stations or simply points on a track. The values of w_{ts} must be strictly followed. In particular, if $w_{ts} = 0$, train type t may not stop in station s even if doing so would improve the objective value. Finally, stopping on the main line is not allowed. That is, we assume that all train stops occur on sidings.

4.4. Mathematical formulation

In this section we present two mixed integer programming models for train timetabling and train platforming. Although, the first one is a novel model, it is still very difficult to solve due mainly to binary decision variables dimensions. For this reason, the second and more efficient model is developed and will be presented in this chapter.

4.4.1. First mathematical model

We now present the first MILP formulation of the problem. This model gives us a general case in which we have multiple classes of train and multiple platforms at (some) stations as discussed before. The indices, parameters, and decision variables in the mathematical program, and their respective explanations, are given in Table 4.1. The input data consists of 14 primary parameters (S, P, s_p, p_s, T, $oStation_t$, $dStation_t$, $trav_s$, w_{ts}, $dMin_{ts}$, $dMax_t$, $hTrack_s$, $hPlatform_p$, and a_k) and 9 secondary parameters that are derived from the primary parameters. The primary parameters are described in Section 4.1. The decision variables, described later in this section, are $Interval$, D_{ts}, A_{ts}, X_{tp}, Z_{tusij}, and Y_{tusij}. The first three variables take real values, and the last three variables are binary. As mentioned before, $Interval$ is the length in minutes of the dispatching cycle. It is the quantity we seek to minimize. A_{ts} and D_{ts} are the arrival and departure times of the original train of type t at station s. The binary variables X_{tp} indicate which train types are assigned to which platforms. The binary variables Z_{tusij} indicate the sequence of trains on the main line. The binary variable Y_{tusij} indicates the sequence of trains that are assigned to the same platform.

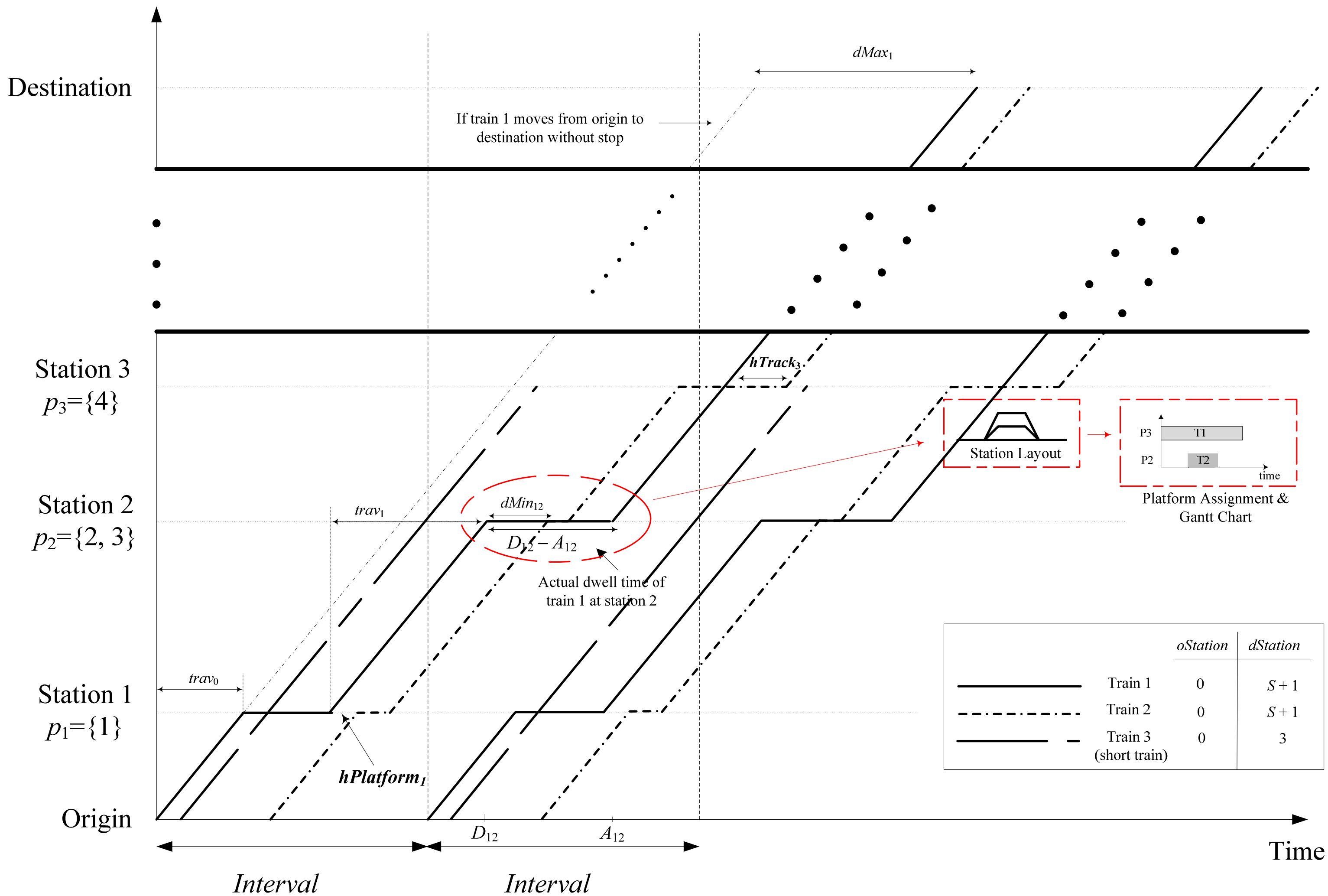


Figure 4.2 Cyclic train timetable depicted as a time-space diagram with input parameters and decision variables indicated.

Table 4.1 Indices, parameters and decision variables in the first mathematical model

Indices	
t, u	Train index
s	Station index
p	Platform index
i, j	Interval index
Parameters	
T	Number of trains
S	Number of intermediate stations
P	Number of platforms ($P \geq S$)
a_1, a_2	Weights for objective function components
$hTrack_s$	Headway on the main line between station s and s+1
$hStation_s$	Headway in station s
$oStation_t$	Origin station for train t
$dStation_t$	Destination station for train t
$trav_s$	Traveling time on the main line from s to s + 1
S_p	Station for platform p
P_s	Set of platforms in station s
$w_{ts}=\begin{cases}1\\0\end{cases}$	If train t stops at station s otherwise
$dMin_{ts}$	Minimum dwell time for train t at station s
$dMax_t$	Maximum dwell time for train t
M	A very large number
$maxIn_{tus}$	Maximum number of intervals for sequence of train t and u at station s
$minIn_{tus}$	Minimum number of interval for sequence of train t and u at station s
$LBInterval_s$	Lower bound on interval at station s
$maxLB$	Maximum lower bound
Decision variables	
$Interval$	Interval duration (real, $\geq$)
A_{ts}	Arrival time of train t at station s, $s : oStation_t < s \leq dStation_t$
D_{ts}	Departure time of train t at station s, $s : oStation_t \leq s < dStation_t$
$X_{tp} = \begin{cases}1\\0\end{cases}$	If train t is assigned to platform p Otherwise
$Y_{tupij} = \begin{cases}1\\0\end{cases}$	If train t of interval i comes before train u of interval j in platform p Otherwise
$Z_{tusij} = \begin{cases}1\\0\end{cases}$	If train t of interval i comes before train u of interval j departing from station s Otherwise

Minimize

$$a_1 \times Interval + a_2 \times \left(\sum_{t=1}^{T} \sum_{s=oStation_t+1}^{dStation_t-1} \left(D_{ts} - A_{ts} \right) \right) \tag{4-1}$$

Subject to

$$A_{t,s+1} = D_{ts} + trav_s \qquad \forall t, \forall s : \text{oStation}_t \le s < dStation_t \qquad (4\text{-}2)$$

$$D_{t,oStation_t} \le Interval + hTrack_s \qquad \forall t, \forall s \in \{0\} \cup \{S\} \qquad (4\text{-}3)$$

$$A_{ts} + dMin_{ts} \le D_{ts} + (1 - w_{ts})M \qquad \forall t, \forall s : oStation_t < s < dStation_t \qquad (4\text{-}4)$$

$$A_{ts} \le D_{ts} \qquad \forall t, \forall s : oStation_t < s < dStation_t \qquad (4\text{-}5)$$

$$A_{ts} \ge D_{ts} - Mw_{ts} \qquad \forall t, \forall s : oStation_t < s < dStation_t \qquad (4\text{-}6)$$

$$\sum_{s=oStation_t+1}^{dStation_t-1} (D_{ts} - A_{ts}) \le dMax_t \qquad \forall t \qquad (4\text{-}7)$$

$$(i \times Interval + D_{ts}) + hTrack_s \le (j \times Interval + D_{us}) + (1 - Z_{tusij})M \qquad (4\text{-}8)$$

$$(j \times Interval + D_{us}) + hTrack_s \le (i \times Interval + D_{ts}) + (1 - Z_{utsji})M \qquad (4\text{-}9)$$

$$Z_{tusij} + Z_{utsji} = 1$$

$$\forall t,u : t \ne u, \forall s : \text{oStation}_t \le s < dStation_t \wedge oStation_u \le s < dStation_u \qquad (4\text{-}10)$$

$$Z_{tusij} \le Z_{t,u,s,i,j+1} \qquad (4\text{-}11)$$

$$Z_{tusij} \le Z_{t,u,s,i-1,j} \qquad (4\text{-}12)$$

$$\sum_{p \in P_s} X_{tp} = w_{ts} \qquad \forall t, \forall s \qquad (4\text{-}13)$$

$$\begin{aligned} &(i \times Interval + D_{t,s_p}) + hStation_{s_p} \le (j \times Interval + A_{u,s_p}) + (1 - Y_{tupij})M \\ &(i \times Interval + A_{t,s_p}) - hStation_{s_p} \ge (j \times Interval + D_{u,s_p}) - Y_{tupij}M \end{aligned} \qquad (4\text{-}14)$$

$$Y_{tupij} \le Y_{t,u,p,i,j+1} \qquad (4\text{-}15)$$

$$Y_{tupij} \le Y_{t,u,p,i-1,j} \qquad (4\text{-}16)$$

$$X_{tp} + X_{up} - 1 \le Y_{tupij} + Y_{utpji} \tag{4-17}$$

$$X_{tp} \ge Y_{tupij} + Y_{utpji} \tag{4-18}$$

$$X_{up} \ge Y_{tupij} + Y_{utpji} \tag{4-19}$$

$$Y_{tupij} \le Z_{t,u,s_p,i,j} \tag{4-20}$$

$$D_{t,s_p} - A_{t,s_p} \le Interval - hStation_{s_p} \qquad \forall t, \forall s, \forall p \tag{4-21}$$

$$maxLB \le Interval \le minUB \tag{4-22}$$

In the mathematical model above, the objective function (4-1) minimizes dispatching interval and total dwell time incurred by all train types running over this single-line with multiple platform railway system combined. The first objective is the main focus of this study, so $a_1 >> a_2$ in most experiments in this study. The constraint (4-2) guarantees that the time between departure and arrival to the next station for a train equals to the traveling time on that link. Constraint (4-3) says that each train is dispatched form its origin within the interval length because we assume that only one train of each type is dispatched per cycle. Constraints (4-4) – (4-6) are defining whether or not a train stops at a particular station, and if so it must stop there for at least a minimum amount time. Constraint (4-7) imposes an upper bound on total actual dwell time of a train. As mentioned since in this problem we consider passing situation that happens only at stations and/or sidings, a set of constraints must be defined to guarantee these kinds of possibilities. These constraints, previously introduced, are called ordering constraints. We have two kinds of ordering constrains. One for every pair of train on the main line and one for every pair of trains stop at the same platform of a particular station. Constraints

(4-8) – (4-10) determine order of two train types on the main line leaving station *s* toward station *s*+1. Constrains (4-11) and (4-12) are logical constraints and relate each train type to its immediate interval successor and interval predecessor. Constraint (4-13) plays the assignment role in this model. According to this equation, each train should assign to exactly one platform only if that station is part of train line plan and the train should shop in that station. Constraint (4-14) is the second ordering constraint as mentioned before. According to this constraint, if two train types are assumed to stop at the same station and are assigned to the same platform, the following train enters the station only after headway on station minutes that the preceding train has departed from that platform. In other words, this constraint guarantees that the platform is empty before to be assigned to a train. Constraints (4-15) and (4-16) are playing the same role at stations as constraints (4-11) and (412) do on the main line. The order of two trains must be determined, only if their assigned to the same platform, and this is achieved via constraints (4-17) – (4-19). According to these constraints, if two train types are assigned to the same platform one should be scheduled after the other. Constraint (4-20) connects two different ordering constraints. In other words, if one train precedes another one at a platform to which both are assigned, the second train type must follow the first train type on the main line following that station. Constraint (4-21) restricts actual dwell time of each train at each station meaning that each train cannot dwell at a station more than interval length. This guarantees that the train does not interfere with the first train of the next cycle which is more important at stations where only one train stops in each cycle. Constraint (4-22) defines the lower bound and upper bound on *Interval* which will be discussed later.

Special cases

In the first mathematical model it is assumed that each train type has different origin and destination. One special case is that all trains are dispatched from the very first station known as origin and should stop at the very last station. In this case, constraint (4-3) is slightly changed into the following form:

$$D_{t,0} \leq Interval - hTrack_0 \qquad \forall t \in T \qquad (4\text{-}23)$$

According to this constraint each train should be dispatched sometime within the interval. The reason of subtracting *hTrack* is to make sure that the last train dispatched is at least separated by headway from the first train in the second cycle. The second special case is the one where each station has exactly one platform or siding for transferring passenger and/or passing other train from the station. In this case, assignment and other related constraints are not required to be defined explicitly. This will reduce the number of binary decision variables and constraints which will speed up computing time.

4.4.2. Second mathematical model: an improvement

The mathematical model proposed in Section 4.4.1 is very difficult to solve since binary decision variables used for ordering trains have five indices which require the model to define a huge amount of decision variables as well as constraint. For instance for a typical rail line with 30 stations, 5 train types and 5 intervals for each trains we need to define 15000 binary variables in order to define ordering on the main track. Added to this consider ordering decisions at platforms, assigning trains to platform and arrival/departure decision variables. For this reason another mixed integer linear program with fewer binary decision variables is defined and presented here. Like the first model the indices, parameters, and decision variables in the mathematical program, and their respective explanations, are given in Table 4.2. The input data consists of 14 primary parameters (S, P, s_p, p_s, T, $oStation_t$, $dStation_t$, $trav_s$, w_{ts}, $dMin_{ts}$, $dMax_t$, $hTrack_s$, $hPlatform_p$, and a_k) and nine secondary parameters that are derived from the primary parameters. The primary parameters are described in Section 4.1. The decision variables, described later in this section, are $Interval$, D_{ts}, A_{ts}, X_{tp}, Z_{itsu}, Y_{itpu}, and V_{itpu}. As before, the first three variables take real values, and the last four variables are binary. A_{ts} and D_{ts} are the arrival and departure times of the original train of type t at station s. The binary variables are the same as defined for the first mathematical model, except the fact that for ordering decision variables (Z, Y) we consider four indices instead of five indices, and we define another ordering decision variable V with four indices. The idea came from the cyclic structure of the model. In other words, because of the cyclic nature of the model, when the origin train of type t is compared with other trains, the same structure will be carried over for the second train of type t, the third train of type t, and so forth.

Table 4.2 Indices, parameters, and decision variables in the second mathematical model

Indices	
s	Station index ($0 \le s \le S$+1; 0 and S+1 represent the origin and destination)
p	Platform index ($1 \le p \le P$)
t, u	Train type index ($1 \le t, u \le T$)
i	Interval index; index for different trains of same type (integer; 0 = original interval)
k	Objective function component (k = 1, 2)
Parameters	
S	Number of intermediate stations (integer, > 0)
P	Number of platforms (integer, $P \ge S$)
s_p	Station in which platform p resides (p = 1 to P)
p_s	Set of platforms in station s (s = 1 to S)
T	Number of train types (integer, > 0)
$oStation_t$	Origin station for train type t (integer, ≥ 0, $\le S$) (t = 1 to T)
$dStation_t$	Destination station for train type t (integer, ≥ 1, $\le S$+1) (t = 1 to T)
$trav_s$	Traveling time on main line from station s to s + 1 (minutes) (real, > 0) (s = 0 to S)
$w_{ts} = \begin{cases} 1 \\ 0 \end{cases}$	If train type t stops at station s otherwise (binary) (t = 1 to T, s = 1 to S)
$dMin_{ts}$	Minimum dwell time for train type t at station s (real, ≥ 0) ($\forall t, \forall s : oStation_t < s < dStation_t$)
$dMax_t$	Maximum allowed total dwell time for train type t at all stations (minutes) (real, ≥ 0) (t = 1 to T)
$hTrack_s$	Headway on the main line between station s and s+1 (minutes) (real, > 0) (s = 0 to S)
$hPlatform_p$	Headway on platform p (minutes) (real, ≥ 0) (p = 1 to P)
a_k	Weight for objective function component k (real, ≥ 0) (k = 1, 2)
$minUB$	Minimum upper bound on optimal value of *Interval* (real, > 0)
$mainLB$	$= \max_s \ (s = 0 \text{ to } S) \left\{ hTrack_s * \left(\sum_{t=1}^{T} (oStation_t \le s < dStation_t) \right) \right\}$ (1=true, 0=false)
$stationLB_s$	Lower bound on *Interval* obtained by assigning train types to platforms in station s (s = 1 to S)
$stationLB$	$= \max_s$ (s = 1 to S) $\{stationLB_s\}$
$maxLB$	Maximum lower bound on optimal value of *Interval* (= max{*mainLB*, *stationLB*}) (real, > 0)
$lowIntMain_{tus}$	Lowest interval of train type t that must be compared to the "interval 0 train of type u" to ensure headway constraints are enforced on the portion of the main line between stations s and s+1 (integer) (defined $\forall$ (t,u): $t < u$, $\forall$ s from 0 to S)

Table 4.2 Continued

$highIntMain_{tus}$	Highest interval of train type t that must be compared to the "interval 0 train of type u" to ensure headway constraints are enforced on the portion of the main line between stations s and s+1 (integer) (defined $\forall$ (t,u): $t < u$, $\forall$ s from 0 to S)
$lowIntPlat_{tup}$	Lowest interval of train type t that must be compared to "interval 0 train of type u" to ensure headway constraints are enforced on platform p (integer) (defined $\forall$ (t,u): $t < u$, $\forall$ p)
$highIntPlat_{tup}$	Highest interval of train type t that must be compared to "interval 0 train of type u" to ensure headway constraints are enforced on platform p (integer) (defined $\forall$ (t,u): $t < u$, $\forall$ p)
Decision variables	
Interval	Interval duration (minutes) (real, > 0)
D_{ts}	Departure time of original train of type t from station s (real, $\geq$0) ($\forall$ t, $\forall$ s: $oStation_t \leq s < dStation_t$)
A_{ts}	Arrival time of original train of type t at station s (real, > 0) ($\forall$ t, $\forall$ s: $oStation_t < s \leq dStation_t$)
$X_{tp} = \begin{cases} 1 \\ 0 \end{cases}$	If train type t is assigned to platform p Otherwise (binary) (t = 1 to T, p = 1 to P)
$Z_{itsu} = \begin{cases} 1 \\ 0 \end{cases}$	If "interval i train of type t" reaches merge point just after station s **before** original (i.e. interval 0) train of type u reaches there Otherwise (binary) ($\forall$ s: $0 \leq s \leq S$, $\forall$ $(t,u$: $t < u)$, $\forall$ i: $lowIntMain_{tus} \leq i \leq highIntMain_{tus}$)
$Y_{itpu} = \begin{cases} 1 \\ 0 \end{cases}$	If "interval i train of type t" uses platform p **before** original (i.e. interval 0) train of type u Otherwise (binary) ($\forall$ p, $\forall$ $(t,u$: $t < u)$, $\forall$ i: $lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$)
$V_{itpu} = \begin{cases} 1 \\ 0 \end{cases}$	If "interval i train of type t" uses platform p **after** original (i.e. interval 0) train of type u Otherwise (binary) ($\forall$ p, $\forall$ $(t,u$: $t < u)$, $\forall$ i: $lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$)

The second MILP formulation of this problem is as follows:

Minimize

$$a_1 \times Interval + a_2 \left(\sum_{t=1}^{T} \sum_{s=oStation_t+1}^{dStation_t-1} \left(D_{ts} - A_{ts} \right) \right) \tag{4-24}$$

Subject to

$$0 \le D_{t,oStation_t} \le Interval \qquad \forall t \in T \tag{4-25}$$

$$A_{t,s+1} = D_{ts} + trav_s \qquad \forall t, \forall s : oStation_t \le s < dStation_t \tag{4-26}$$

$$A_{ts} + dMin_{ts} \le D_{ts} + \left(1 - w_{ts}\right) M \qquad \forall t, \forall s : oStation_t < s < dStation_t \tag{4-27}$$

$$A_{ts} \le D_{ts} \qquad \forall t, \forall s : oStation_t < s < dStation_t \tag{4-28}$$

$$A_{ts} \ge D_{ts} - Mw_{ts} \qquad \forall t, \forall s : oStation_t < s < dStation_t \tag{4-29}$$

$$\sum_{s=oStation_t+1}^{dStation_t-1} \left(D_{ts} - A_{ts} \right) \le dMax_t \qquad \forall t \in T \tag{4-30}$$

$$\left(i \times Interval + D_{ts} \right) - D_{us} - \left(1 - Z_{itsu}\right) M \le -hTrack_s \tag{4-31}$$

$$\left(i \times Interval + D_{ts} \right) - D_{us} + Z_{itsu} M \ge hTrack_s \tag{4-32}$$

$$\forall s \in \{0, ..., S\}, \forall \left(t, u : t < u \wedge oStation_t \le s < dStation_t \wedge oStation_u \le s < dStation_u \right)$$

$$\forall i : lowIntMain_{tus} \le i < highIntMain_{tus}$$

$$Z_{itsu} \le Z_{i-1,t,s,u} \tag{4-33}$$

$$\forall s \in \{0, ..., S\}, \forall \left(t, u : t < u \wedge oStation_t \le s < dStation_t \wedge oStation_u \le s < dStation_u \right)$$

$$\forall i : lowIntMain_{tus} \le i < highIntMain_{tus}$$

$$\sum_{p \in P_s} X_{tp} = w_{ts} \qquad \forall t, \forall s \in \{1,...,s\} \tag{4-34}$$

$$X_{tp} + X_{up} - 1 \leq Y_{itpu} + V_{itpu} \tag{4-35}$$

$$\forall p, \forall (t,u : t < u), \forall i : lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$$

$$X_{tp} \geq Y_{itpu} + V_{itpu} \tag{4-36}$$

$$\forall p, \forall (t,u : t < u), \forall i : lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$$

$$X_{up} \geq Y_{itpu} + V_{itpu} \tag{4-37}$$

$$\forall p, \forall (t,u : t < u), \forall i : lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$$

$$A_{u,s_p} - \left(i \times Interval + D_{t,s_p}\right) + \left(1 - Y_{itpu}\right) M \geq hPlatform_p \tag{4-38}$$

$$\left(i \times Interval + A_{t,s_p}\right) - D_{u,s_p} + \left(1 - V_{itpu}\right) M \geq hPlatform_p \tag{4-39}$$

$$\forall p, \forall (t,u : t < u), \forall i : lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$$

$$Y_{itpu} \leq Y_{i-1,t,p,u} \tag{4-40}$$

$$\forall p, \forall (t,u : t < u), \forall i : lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$$

$$V_{itpu} \leq V_{i+1,t,p,u} \tag{4-41}$$

$$\forall p, \forall (t,u : t < u), \forall i : lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$$

$$Y_{itpu} \leq Z_{i,t,s_p,u} \tag{4-42}$$

$$\forall p, \forall (t,u : t < u), \forall i : lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$$

$$V_{itpu} \leq \left(1 - Z_{i,t,s_p,u}\right) \tag{4-43}$$

$$\forall p, \forall (t,u : t < u), \forall i : lowIntPlat_{tup} \leq i \leq highIntPlat_{tup}$$

$$Interval \geq \left(D_{t,s_p} - A_{t,s_p}\right) + \left(X_{tp}\right)\left(Platform_p\right) \tag{4-44}$$

$$\forall t, \forall p : oStation_t \leq s_p \leq dStation_t$$

$$maxLB \leq Interval \leq minUB \tag{4-45}$$

The objective function (4-24) is the same as the first model and has two parts, weighted a_1 and a_2, which respectively pursue the minimization of the cycle length and the minimization of the total delay experienced by all train types at all stations combined. The first objective is the main focus of this study, so $a_1 >> a_2$ in most experiments in this study. Constraint (4-25) ensures that the original train of each type departs its starting point sometime during the first interval (i.e. "interval 0"). Constraint (4-26) requires that $trav_s$ be the traveling time for all train types along the main line between stations s and s+1. Constraints (4-27) – (4-29) are related to train type line plan and ensure that (i) each train type stops for the required minimum amount of time in each of the stations it visits and that (ii) train type t does not spend any time in station s if w_{ts} = 0. Constraint (4-30) ensures that the total dwell time of train type t in all stations combined does not exceed the maximum allowed value $dMax_t$. Constraints (4-31) and (4-32) are disjunctive constraints that enforce the headway restrictions on the portion of the main line between stations s and s+1 for all s from 0 to S. In particular, these constraints guarantee that the interval i train of type t appears at the merge point just after station s either at least $hTrack_s$ minutes before (4-31) or after (4-32) the original train of type u appears there for all s from 0 to S, for all pairs of train types that travel along the portion of the portion of the main line between stations s and s+1, and for all intervals i of train type t that could possibly interfere with the original train of type u along that stretch of the main line.

Note that, although constraints (4-31) and (4-32) consider many trains of type *t* and only one train of type *u*, they enforce headway constraints on the main track for *all* trains of type *t* versus *all* trains of type *u* owing to the repetitive, cyclic nature of the timetable. Regarding the ordering of trains on the main line, constraint (4-33) states that if the interval *i* train of type *t* is before the original train of type *u*, then the interval *i*-1 train of type *t* must also be before the original train of type *u*. Constraint (4-34) ensures that each train type stopping (not stopping) in a station visits 1 (0) platform(s) in the station. Constraints (4-35) – (4-37) ensure that, if two train types utilize the same platform, then the first train type must either use the platform before or after the second train type; the two train types cannot utilize the platform simultaneously. Constraints (4-38) – (4-39) enforce the headway restrictions on all station platforms. In particular, if both train types *t* and *u* use platform *p*, these constraints guarantee that the interval *i* train of type *t* uses platform *p* either at least $hPlatform_p$ minutes before (4-38) or after (4-39) the original train of type *u* uses it for all *p* and for all intervals *i* of train type *t* that could possibly interfere with the original train of type *u* at that platform. Note that, although constraints (4-38) – (4-39) consider many trains of type *t* and only one train of type *u*, they enforce headway constraints on the station platforms for *all* trains of type *t* versus *all* trains of type *u* owing to the repetitive, cyclic nature of the timetable. Regarding the ordering of trains on the platforms, constraints (4-40) – (4-41) state that if the interval *i* train of type *t* is before (after) the original train of type *u*, then the interval *i*-1 (*i*+1) train of type *t* must also be before (after) the original train of type *u*. Constraints (4-42) – (4-43) ensure that the ordering of two trains on the main line between stations *s* and *s*+1 agrees with the ordering in which these trains visit the same platform in station *s*. Constraint (4-44) ensures that the cycle length *Interval* is large enough so that each stop by a train at a platform can be made without the train overlapping with its sister train from the next interval. Constraint (4-45) forces *Interval* to take a value that is no lower than its lower bound and no higher than its upper bound. These constraints are redundant but help to shorten the time required to find an optimal solution.

4.4.3. Upper bound on optimal value

The parameter *minUB* provides an upper bound on the optimal value of *Interval*. It equals the lowest value of *Interval* among the (*T*-1)! feasible solutions that are formed by considering all possible (*T*-1)! cyclic orderings of the train types and then scheduling train types in order one-at-a-time such that (A) all train types achieve their minimum station dwell times; (B) there is no passing; and (C) no two train types may be in the same station at the same time. During the above scheduling process, the first train type is scheduled so it departs its starting point at time 0 and achieves its minimum dwell times in all stations it visits. Each subsequent train type is scheduled one-at-a-time by (1) assuming it departs its starting point at a very large time value (e.g. 1000) and achieves its minimum station dwell times in all stations it visits and then (2) repeatedly scheduling the train's departure earlier (i.e. pushing the train to the left) until it is as close as possible to the already scheduled train types without violating main line headway and station (i.e. platform) headway constraints. During the above scheduling process, we forbid the overlapping of trains in stations. In the example, the train type ordering 2-1-3 produces the best upper bound of 17 (Figure 4.3 part B).

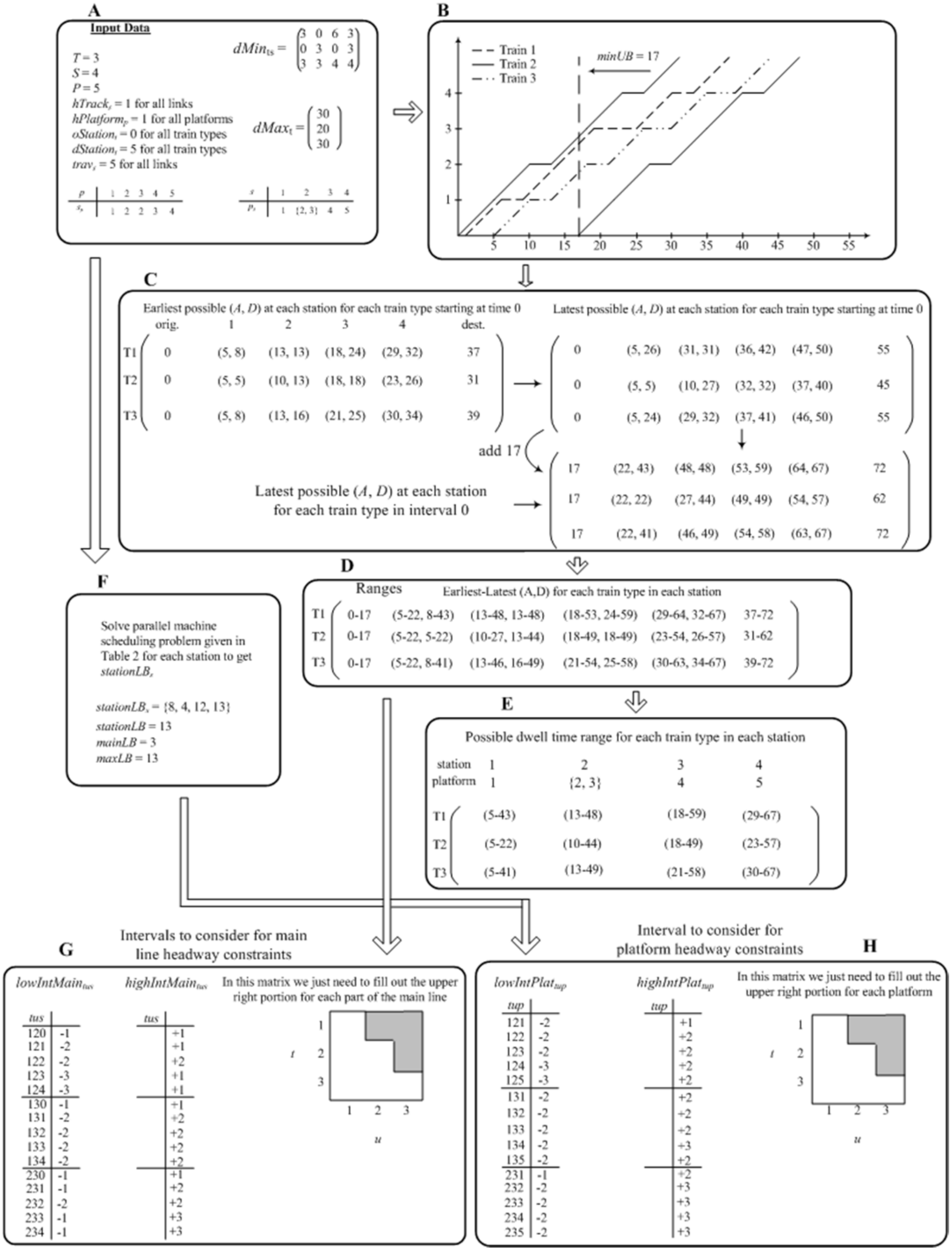


Figure 4.3The procedure for determining secondary parameters

4.4.4. Lower bound on optimal value

The parameter *maxLB* provides a lower bound on the optimal value of *Interval*. It equals the higher of two different lower bounds—*mainLB* and *stationLB*—that are computed as follows. Parameter *mainLB* is the smallest value of *Interval* that could possibly be achieved due to headway constraints on the main line only. It equals the highest value of ($hTrack_s$) × (number of train types that run on the portion of the main line between stations s and s+1) among all s from 0 to S. Parameter *stationLB* is the smallest value of *Interval* that could possibly be achieved due to dwell time, headway, and capacity constraints inside the stations only. It is obtained by solving, for each station s, the simple parallel machine scheduling problem shown in Table 4.3. The problem is formed by considering each platform p in the station as a machine, each train type t that stops in station s as a job with duration $dMin_{ts}$, and $hPlatform_p$ to be the setup time prior to each job that is scheduled on platform p. The objective of the machine scheduling problem is to minimize the makespan. Let $stationLB_s$ be the optimal value obtained by solving the machine scheduling problem for station s. Then $stationLB = \max_s\{stationLB_s\}$. The values of *mainLB*, $stationLB_s$ for all s, *stationLB*, and *maxLB* are shown in Figure 4.3 part C.

4.4.5. Derivation of secondary parameters

We now describe the secondary parameters. Figure 4.3 illustrates how several of the secondary parameters are computed for a small problem instance. The primary parameters for the problem instance are displayed in Figure 4.3 part A. Note that w_{ts} = 0 (1) when $dMin_{ts}$ is zero (nonzero).

The secondary parameters $lowIntMain_{tus}$, $highIntMain_{tus}$, $lowIntPlat_{tup}$, and $highIntPlat_{tup}$ allow us to construct constraints enforcing the headway restrictions on each portion of the main line and on each platform. Due to the cyclic nature of the timetable, these constraints often need to be investigated not only for the *original train* (i.e. *interval 0 train*) of each type (that departs its starting point sometime between time 0 and *Interval*) but also to other trains of the same type that depart their starting point in intervals before or after the original one. Here we define the *interval i train of type t* to be the train of type t that departs its starting point during interval i, i.e. sometime between time $(i) \times Interval$ and time $(i+1) \times Interval$ (i is any negative or non-negative integer). In general, the above parameters indicate how many "intervals worth" of trains of a given type could possibly have a headway conflict with another train type on a certain station platform or along a certain portion of the main line. In particular, the parameters $lowIntPlat_{tup}$ and $highIntPlat_{tup}$ ($lowIntMain_{tus}$ and $highIntMain_{tus}$) indicate the lowest and highest interval trains of type t that could possibly come within $hPlatform_p$ ($hTrack_s$) minutes of the original train of type u on platform p (the portion of the main line between stations s and s+1).

Table 4.3 Parallel machine scheduling problem for computing *stationLB*$_s$

Indices	
T	train type
P	platform
Parameters	
S	Station under consideration.
T	Number of train types stopping in station *s*
P	Number of platforms in station *s*
$dMin_t$	Minimum required dwell time for train type *t* in the station (= $dMin_{ts}$ in Table 4.2, real, > 0)
$hPlatform_p$	Minimum required headway between trains stopping on platform *p* (real, ≥ 0)
Decision variables	
X_{tp}	= 1 if train type *t* is assigned to platform *p* (binary).
$stationLB_s$	Minimum makespan for the machine scheduling problem (real, > 0).
Math program	

Objective:

$$\text{minimize} \quad stationLB_s$$

Subject to:

$$\sum_{p=1}^{P} X_{tp} = 1 \quad \forall t$$

$$\sum_{t=1}^{T} (X_{tp})(dMin_t + hPlatform_p) \le stationLB_s \quad \forall p$$

The computation of parameters $lowIntMain_{tus}$, $highIntMain_{tus}$, $lowIntPlat_{tup}$, and $highIntPlat_{tup}$ begins with a calculation of the earliest and latest possible times when the original train of each type could possibly *arrive at* and *depart from* each station (including departing from the origin and arriving at the destination) (Figure 4.3 part E). The earliest possible arrival and departure times are calculated in a straightforward manner (part D upper left). The latest possible arrival and departure times are calculated by first assuming that each train type t departs at time 0 and spends the maximum total dwell time $dMax_t$ in stations; backwards recursion is used to make the arrival and departure times as high as possible while still adhering to main line traveling times and minimum required station dwell times (part D upper right). These values are then increased by *minUB* (part D bottom right). Part E is taken from the upper left and bottom right portions of part D. Figure 4.3 part F shows the possible times when the original train of each type can dwell (i.e. stay, be present) in each intermediate station given the information in part E.

The calculation of $lowIntMain_{tus}$ and $highIntMain_{tus}$ is shown in Figure 4.3 part G. This calculation makes use of information in parts C and E. For each pair of train types t-u, we determine the lowest and highest numbered interval of train type t that could still interfere with (i.e. be strictly less than $hTrack_s$ minutes away from) the original train of type u on each portion s of the main line ($0 \leq s \leq S$). Consider the computation for $(t,u,s) = (1,2,1)$. Here we refer to part E and we compare the feasible time window for the original train of type 1's departure from station 1 (8-43) to the feasible time window for the original train of type 2's departure from station 1 (5-22) and we add or subtract multiples of $maxLB = 13$ to the former time window until it is separated from the latter time window by at least $hTrack_1$ = 1. Subtracting 1×13, 2×13, and 3×13 from (8-43) yields ($^{-}$5-30), ($^{-}$18-17), and ($^{-}$31-4) respectively. The first two time windows are within $hTrack_1$ = 1 of time window (5-22) (they overlap with this window) but the third time window is at least $hTrack_1$ = 1 minutes away from the window (5-22) (the windows are exactly 1 minute apart). Thus, the original, interval -1, or interval -2 train of type 1 might interfere with the original train of type 2 on the portion of the main line between stations 1 and 2. But it is not possible for the interval -3 train of type 1 to interfere with the original train of type 2 on the portion of the main line between stations 1 and 2. In other words, $lowIntMain_{121}$ = -2. Adding 1×13 and 2×13 to the window (8-43) yields (21-56) and (34-69) respectively. The first window is within $hTrack_1$ = 1 of time window (5-22) but the second time window is at least $hTrack_1$ = 1 minutes away from the time window (5-22). Thus, $highIntMain_{121}$ = +1.

The calculation of $lowIntPlat_{tup}$ and $highIntPlat_{tup}$ is shown in Figure 4.3 part H. This calculation is similar to that for $lowIntMain_{tus}$ and $highIntMain_{tus}$ except that we refer to part F instead of part E. Here, for each pair of train types t-u, we determine the lowest and highest numbered interval of train type t that could still interfere with (i.e. be strictly less than $hPlatform_p$ minutes away from) the original train of type u on each platform p ($1 \leq p \leq P$) that is eligible to be visited by both train types. The term “n/a” indicates that two train types cannot possibly visit a platform.

4.4.6. What is the best value for Big M?

Another parameter that needs to be determined is M in the mixed integer linear program. In all mixed integer programs this value is defined as an arbitrary number, but care must be taken in determining and using this value. As one of the characteristic of a good model is the tightness of constraints, it is important what value is picked up for M in the formulation in such a way that it guarantees feasibility of the model, because picking up a large value (compared to the other parameters) may result is infeasibility of the solution. For this reason, a simple method based on the disjunctive constraints in the model (referred to as ordering constraints) will be presented in this section. Without loss of generality, let us consider constraint (4-31) again for calculation purposes. The other two ordering constraints can be used as well.

$$\left(i \times Interval + D_{ts}\right) - D_{us} - \left(1 - Z_{itsu}\right)M \leq -hTrack_s \qquad (4\text{-}31)$$

For this purpose, without loss of generality, we assume that one train moves as fast as possible and the other one moves as slow as possible. Further we will assume that this constraint must be satisfied for all station and in order for we can better differentiate between these two trains, the last station is considered for our calculation. Therefore, (4-31) would be

$$\left(i \times Interval + D_{tS}\right) - D_{uS} + Z_{itSu}M \leq M - hTrack_s \qquad (4\text{-}31^*)$$

Suppose train u is the fast train and train t is the slow train. Further assume that Z_{itSu} = 0. In order to estimate a value for M, we need to use other variables' estimation. Also, to make M as large as possible the first parenthesis should get the largest value possible and the second parenthesis should get the smallest value possible. Since train of type t is slow, it should stop for maximum possible time which is maximum dwell time and largest value for $i = max\ \{highIntPlat_{tup}, highIntMain_{tus}\}$, $\forall t,u \in T : t \neq u$ for all t and u. And since train of type u is fast, it should stop at each station for minimum possible amount of time which is minimum dwell time and $j = 0$, the fast train in the first interval. By substituting these estimates in (4-31*) we have

$$\left(max\left(highIntPlat_{tup}, highIntMain_{tus}\right) \times maxLB + dMax_t\right) - \sum_{s=1}^{S} dMin_{ts} \approx M \qquad (4\text{-}46)$$

4.4.7. Problem complexity

We now prove a result regarding the complexity of the problem at hand.

Theorem 1. The optimization problem at hand—defined by the MILP formulation in Section 3.2—is *NP*-hard.

Proof. Consider the decision problem corresponding to the optimization problem *P*||*Cmax*, i.e. the *P*-machine parallel machine scheduling problem without preemption where the objective is to minimize the makespan. This decision problem is known to be *NP*-complete (Garey and Johnson (1978)). Furthermore, problem *P*||*Cmax* is polynomially reducible to the problem at hand. Indeed, problem *P*||*Cmax* is identical to an instance of the problem at hand where $S = 1$; $P = P$ (the number of platforms in the train problem equals the number of machines in the machine scheduling problem); T equals the number of jobs in problem *P*||*Cmax*; $oStation_t = 0$ for all t; $dStation_t = 2$ for all t; $trav_s = 0$ for $s = 0$ and 1; $w_{t1} = 1$ for all t; $dMin_{t1}$ equals the processing time of job t in problem *P*||*Cmax* for all t; $dMax_t = dMin_{t1}$ for all t; $hTrack_s = 0$ for $s = 0$ and 1; $hPlatform_p = 0$ for all p; $a_1 = 1$; and $a_2 = 0$. In other words, the task of finding a cyclic platform schedule with the minimum possible *Interval* at a single train station having *P* platforms and *T* trains stopping per cycle (with no meaningful constraints other than those that prevent train overlapping on platforms) is equivalent to finding the minimum possible makespan in a nonpreemptive parallel machine scheduling problem with *P* machines and T jobs. Since problem $P||C_{max}$ is polynomially reducible to the problem at hand and decision problem $P||C_{max}$ is NP-complete, it follows that the decision problem at hand is also NP-complete. That is, the optimization problem at hand—defined by the MILP formulation in Section 4.4—is NP-hard. ■

4.5. Illustrative examples

The second mathematical formulation was coded into Microsoft Visual C++ 2010. IBM ILOG Concert Technology was used to define the model within C++ and call the MILP solver IBM ILOG CPLEX 11.2 to solve instances defined in text files. The code includes procedures for automatically computing the secondary parameters $stationLB_s$, $stationLB$, $maxLB$, $lowIntMain_{tus}$, $highIntMain_{tus}$, $lowIntPlat_{tup}$, and $highIntPlat_{tup}$, then the $minUB$ and lower bounds are calculated before the constraints are constructed. The computation of $stationLB_s$ for all s involves calling CPLEX to solve several small machine scheduling problems—one in each station. Two IBM-compatible computers were used to solve the problem instances considered in this study. The first was a desktop computer running Windows XP with two 2.83 GHz processors and 2 GB of RAM. The second was a laptop computer running Windows 7 with eight 1.73 GHz processors and 6 GB of RAM.

Table 4.4 Input data for illustrative example #1

T	S	P	$hTrack_s$	$hPlatform_p$	a_1	a_2	$oStation_t$	$dStation_t$
4	12	14	2 for all s	1 for all p	1	0.0001	0 for all t	13 for all t

	Stat. 1	Stat. 2	Stat. 3	Stat. 4	Stat. 5	Stat. 6	Stat. 7	Stat. 8	Stat. 9	Stat. 10	Stat. 11	Stat. 12	Destination	$dMax_t$
#Platforms	1	1	1	2	1	1	1	2	1	1	1	1	-	
$dMin_{1s}$	0	0	0	0	0	0	0	0	0	0	0	0	-	0
$dMin_{2s}$	0	0	0	3	0	0	0	4	0	0	0	0	-	20.5
$dMin_{3s}$	0	3	0	4	0	5	0	4	0	2	0	3	-	41.5
$dMin_{4s}$	3	5	1	5	3	4	3	1	4	6	2	4	-	70.5
$trav_{s-1}$	7	5	5	9	6	8	9	7	6	8	5	7	5	

$stationLB$ = max {4, 10, 2, 9, 4, 11, 4, 7, 5, 10, 3, 9} = 11 $\quad$ $maxLB$ = max{(4*2), 11} = 11 $\quad$ $minUB$ = 49

$lowIntMain_{14s}$

tus	
1,4,0	-4
1,4,1	-4
1,4,2	-3
1,4,3	-3
1,4,4	-3
1,4,5	-3
1,4,6	-2
1,4,7	-2
1,4,8	-2
1,4,9	-1
1,4,10	-1
1,4,11	-1
1,4,12	0

$highIntMain_{14s}$

tus	
1,4,0	+4
1,4,1	+7
1,4,2	+8
1,4,3	+8
1,4,4	+8
1,4,5	+8
1,4,6	+9
1,4,7	+9
1,4,8	+9
1,4,9	+9
1,4,10	+10
1,4,11	+10
1,4,12	+11

$lowIntPlat_{34p}$

tup	
3,4,1	n/a
3,4,2	-6
3,4,3	n/a
3,4,4	-6
3,4,5	-6
3,4,6	n/a
3,4,7	-5
3,4,8	n/a
3,4,9	-5
3,4,10	-5
3,4,11	n/a
3,4,12	-5
3,4,13	n/a
3,4,14	-4

$highIntPlat_{34p}$

tup	
3,4,1	n/a
3,4,2	+7
3,4,3	n/a
3,4,4	+8
3,4,5	+8
3,4,6	n/a
3,4,7	+8
3,4,8	n/a
3,4,9	+8
3,4,10	+8
3,4,11	n/a
3,4,12	+8
3,4,13	n/a
3,4,14	+9

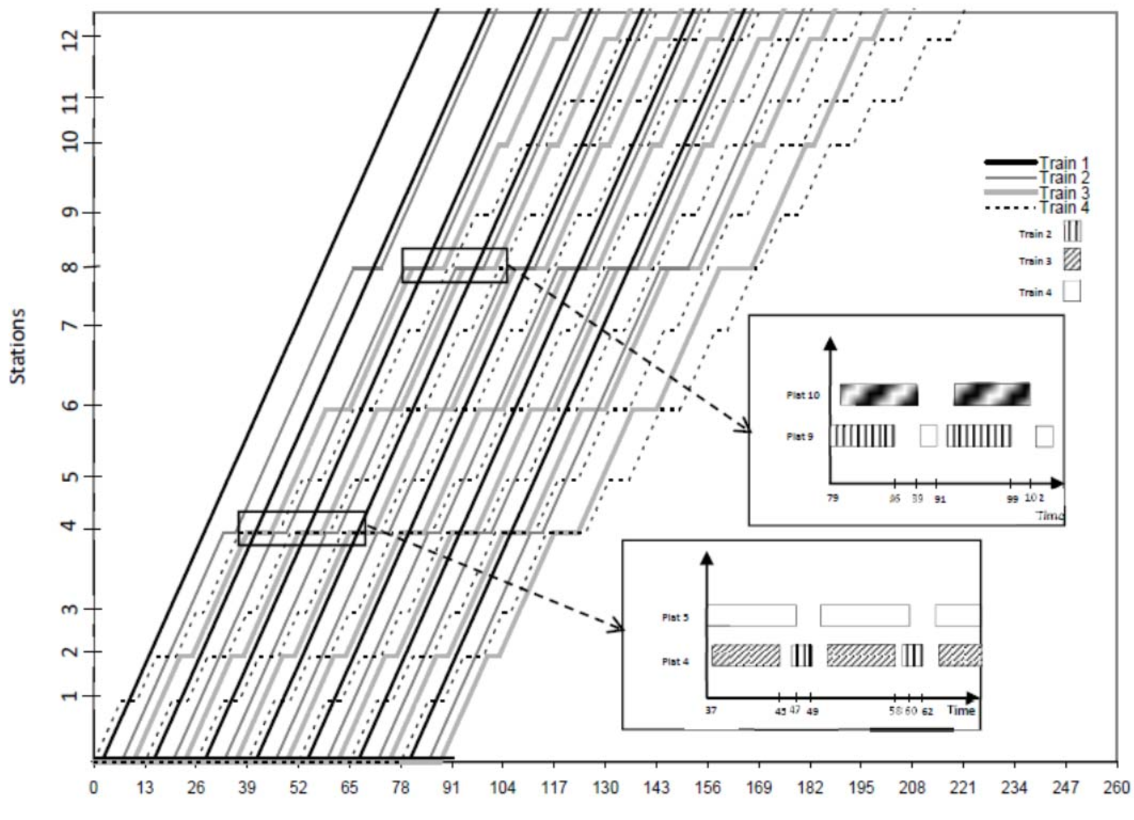
We now present four illustrative examples and discuss their optimal solutions. Before doing so, we first discuss the setup common to all problem instances discussed in this section and Section 4.6. This setup reflects the fact that the primary objective in the math model is to minimize the cycle time and the secondary, subordinate objective is to minimize the total train delay. To accomplish this, all primary parameters besides a_1 and a_2 are multiples of 0.5. This ensures that the optimal value of *Interval* is also a multiple of 0.5. Also, $a_1 = 1$, $a_2 = .0001$, and $\sum_{t=1}^{T} dMax_t \leq 4999.5$. Thus, 0.49995 is the maximum possible value of the second portion of the objective function, and 0.5 is the minimum change in the value of the first portion of the objective function when *Interval* changes. Thus, the weight for a_2 is small enough so that it does not interfere with the primary goal of minimizing the cycle length but is large enough to be able to identify, among all solutions tied for having the minimum *Interval*, a solution that ties for having the smallest total dwell time for all train types at all stations combined. Unless otherwise mentioned, the input parameters throughout Section 4 are as follows: $hPlatform_p = 1$ for all p, and $oStation_t = 0$ and $dStation_t = S+1$ for all t.

4.5.1. First illustrative example

The input data for the first problem instance is presented in Table 4.4. This problem instance considers four train types with different stopping frequencies. Train types 1, 2, 3, and 4 are super express, limited express, express, and local trains respectively. The primary parameters are shown on the top of Table 4.4. The parameters s_p and p_s are not explicitly stated but can be inferred from row "#Platforms" which gives the number of platforms in each station. Also, we assume that $w_{ts} = 0$ (1) when $dMin_{ts}$ is zero (nonzero). The secondary parameters are shown on the bottom of Table 4.4. Due to space limitations, only a subset of the values of $lowIntMain_{tus}$, $highIntMain_{tus}$, $lowIntPlat_{tup}$, and $highIntPlat_{tup}$ are displayed.

Table 4.5 shows an optimal solution for this instance in the form of a timetable. This solution was obtained in 200 seconds. The platform assignments are shown at the top; actual dwell time in each station for each train type is shown in the middle; and detailed schedules for the first five trains of each type are displayed at the bottom. The schedule of each train is fully defined by the arrival and departure times of the train at each station. Figure 4.4 shows the same solution displayed in the form of a time-space diagram. The diagram displays the progression of the first seven trains of each type from the origin to destination. The platform schedules for stations 4 and 8 are shown in the bottom right of the figure. In these platform schedules, the train departure times are labeled but the train arrival times are not.

As Table 4.5 and Figure 4.4 indicate, the optimal value for this instance—the minimum value of *Interval*—is 13 minutes. This value is strictly greater than the lower bound *maxLB* (= 11) and strictly less than the upper bound *minUB* (= 49). Thus, the optimal value is probably not obtainable using a simple procedure such as trial-and-error, and it appears that math programming is the method most suited to address this NP-hard problem.



Time (sec)

Figure 4.4 Optimal time-space diagram for illustrative example #1.

Table 4.5 Optimal track assignment and timetable for illustrative example #1

Minimum *Interval* = 13		Stat. 1	Stat. 2	Stat. 3	Stat. 4		Stat. 5	Stat. 6	Stat. 7	Stat. 8		Stat. 9	Stat. 10	Stat. 11	Stat. 12
		Plat. 1	Plat. 2	Plat. 3	Plat. 4	Plat. 5	Plat. 6	Plat. 7	Plat. 8	Plat. 9	Plat. 10	Plat. 11	Plat. 12	Plat. 13	Plat. 14
Platform Assignment	Train 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Train 2	0	0	0	1	0	0	0	0	1	0	0	0	0	0
	Train 3	0	1	0	1	0	0	1	0	0	1	0	1	0	1
	Train 4	1	1	1	0	1	1	1	1	1	0	1	1	1	1

Actual station dwell times

	Stat. 1	Stat. 2	Stat. 3	Stat. 4	Stat. 5	Stat. 6	Stat. 7	Stat. 8	Stat. 9	Stat. 10	Stat. 11	Stat. 12	Total
Train 1	0	0	0	0	0	0	0	0	0	0	0	0	0
Train 2	0	0	0	3	0	0	0	7	0	0	0	0	10
Train 3	0	3	0	6	0	6	0	8	0	2	0	3	28
Train 4	4	5	2	10	5	5	3	1	4	6	7	6	58

Timetable

	Origin	Stat. 1	Stat. 2	Stat. 3	Stat. 4	Stat. 5	Stat. 6	Stat. 7	Stat. 8	Stat. 9	Stat. 10	Stat. 11	Stat. 12	Destination
Train 1	2	9	14	19	28	34	42	51	58	64	72	77	84	89
Train 2	7	14	19	24	33-36	42	50	59	66-73	79	87	92	99	104
Train 3	10	17	22-25	30	39-45	51	59-65	74	81-89	95	103-105	110	117-120	125
Train 4	0	7-11	16-21	26-28	37-47	53-58	66-71	80-83	90-91	97-101	109-115	120-127	134-140	145
Train 1	15	22	27	32	41	47	55	64	71	77	85	90	97	102
Train 2	20	27	32	37	46-49	55	63	72	79-86	92	100	105	112	117
Train 3	23	30	35-38	43	52-58	64	72-78	87	94-102	108	116-118	123	130-133	138
Train 4	13	20-24	29-34	39-41	50-60	66-71	79-84	93-96	103-104	110-114	122-128	133-140	147-153	158
Train 1	28	35	40	45	54	60	68	77	84	90	98	103	110	115
Train 2	33	40	45	50	59-62	68	76	85	92-99	105	113	118	125	130
Train 3	36	43	48-51	56	65-71	77	85-91	100	107-115	121	129-131	136	143-146	151
Train 4	26	33-37	42-47	52-54	63-73	79-84	92-97	106-109	116-117	123-127	135-141	146-153	160-166	171
Train 1	41	48	53	58	67	73	81	90	97	103	111	116	123	128
Train 2	46	53	58	63	72-75	81	89	98	105-112	118	126	131	138	143
Train 3	49	56	61-64	69	78-84	90	98-104	113	120-128	134	142-144	149	156-159	164
Train 4	39	46-50	55-60	65-67	76-86	92-97	105-110	119-122	129-130	136-140	148-154	159-166	173-179	184
Train 1	54	61	66	71	80	86	94	103	110	116	124	129	136	141
Train 2	59	66	71	76	85-88	94	102	111	118-125	131	139	144	151	156
Train 3	62	69	74-77	82	91-97	103	111-117	126	133-141	147	155-157	162	169-172	177
Train 4	52	59-63	68-73	78-80	89-99	105-110	118-123	132-135	142-143	149-153	161-167	172-179	186-192	197

The passing structure in the optimal solution is as follows. Train type 1—the super express—passes train types 2, 3, and 4 at the following stations respectively: {8}, {4, 8}, {1, 4, 6, 10, 12}. Train type 2—the limited express—passes train types 3 and 4 at the following stations respectively: {6}, {2, 5, 10, 12}. Train type 3—the express—passes train type 4 at the following stations: {4, 11}. Most passing consists of the faster train type completely bypassing the station where the slower train type is stopped. However, for the case where train type 3 passes train type 4 in station 4, both train types stop in the station but the faster train type arrives later and departs earlier. Note that no passing is observed in stations 3, 7, and 9, indicating that the sidings at these stations would not be necessary if stopping on the main line were allowed. Note that the number of super express trains that have already passed each local train by the time the local train gets to the merge point after station s is 1, 1, 1, 2, 2, 3, 3, 3, 3, 4, 4, and 5 for s = 1 to 12 respectively. These numbers are less than their respective upper bounds—7, 8, 8, 8, 8, 9, 9, 9, 9, 10, 10, and 11 (+ 1)—that come from the secondary parameter $highIntMain_{14s}$. Thus, constraints (4-31) – (4-33) in the second mixed integer program guarantee, beyond a doubt, that enough super express trains have been compared to the interval 0 local train to ensure that this pair of train types does not violate the headway restrictions on any portion of the main line.

Our final observations are as follows. The actual total station dwell times for train types 2, 3, and 4 are 10, 28, and 58 respectively which are strictly greater than the respective minimum required values but strictly less than the maximum allowed values of 20.5, 41.5, and 70.5 respectively. The sum of all station dwell times (= 10 + 28 + 58 = 96) is as small as possible given the restriction that *Interval* be minimized. This value can probably be reduced below 96, but not without increasing *Interval* above 13. Finally, the detailed platform schedules for stations 4 and 8 indicate that both stations are busy with an average platform utilization exceeding 50%. Thus, it appears that two sidings are necessary in both stations in order for *Interval* to achieve the value of 13.

4.5.2. Second illustrative example

The inputs and the results related to the second problem instance are summarized in Figure 4.5. The values of the primary input parameters for this instance are shown at the top of the figure. This problem instance considers four train types, five stations, and 15 platforms—three in each station. Here, every train type stops in every station. Due to space limitations, the secondary parameters are not displayed. An optimal solution for this instance in the form of a time-space diagram is displayed at the bottom of the figure. This solution is obtained in 21 seconds. Note that the minimum cycle length found by CPLEX is 12 minutes. In the diagram, the horizontal lines represent train stops. The platform assignments in each station are indicated by slight differences in the vertical placement of the horizontal lines. Let us assume the platforms are numbered 1-15 from the bottom to the top of the diagram. Then, according to the diagram, the assignment of train types to platforms is as follows. In station 1, train types 1, 2, 3, 4 are assigned to platforms 3, 1, 1, 2 respectively. In station 2, train types 1, 2, 3, 4 are assigned to platforms 4, 6, 5, 4. In station 3, train types 1, 2, 3, 4 are assigned to platforms 7, 8, 8, 9. In station 4, train types 1, 2, 3, 4 are assigned to platforms 11, 12, 10, 10. In station 5, train types 1, 2, 3, 4 are assigned to platforms 13, 14, 15, 13. Although this solution uses all 15 platforms, there exists another solution with the same timetable that uses only two platforms in stations 2, 4, and 5. However, there is no solution with the same timetable that uses two or fewer platforms in station 1 or station 3. A related question is whether three platforms are needed in station 1 and station 3 in order for *Interval* to be 12. The input data $dMin_{t1}$ indicate that three platforms are needed in station 1; the optimizer needs to be re-run to determine if three platforms are needed in station 3. Note that the timetable exhibits the phenomenon of *multiple overtaking* (Burdett and Kozan, 2009) in which one train type passes another train type at one station and is subsequently passed by the same train type at another station. In particular, train type 3 passes train type 4 in station 1, but train type 4 passes train type 3 in station 5. This phenomenon is inevitable when the capacity maximization is of interest.

4.5.3. Third illustrative example: Taiwanese high-speed railway system

Our third illustrative example is taken from the Taiwanese high-speed railway system. The entire system consists of two single-track lines: one running southbound from Taipei (the origin) to Zuoying (the destination) with six intermediate stations and the other running northbound from Zuoying to Taipei with the same intermediate stations (Taiwanese High Speed Rail website, 2012).

T		S		P	
4		5		15	
	$dMin_{1s}$	$dMin_{2s}$	$dMin_{3s}$	$dMin_{4s}$	$trav_s$
Station 1	7	3	5	9	9
Station 2	1	2	3	4	7
Station 3	2	1	3	4	5
Station 4	4	4	5	4	6
Station 5	2	1	8	1	9
			$dMax_t$		
	34.5	20.5	35.5	38	
$hPlatform_p$ = 1 for all p		$hTrack_s$ = 3 for all s		Runtime (min) = 1080	

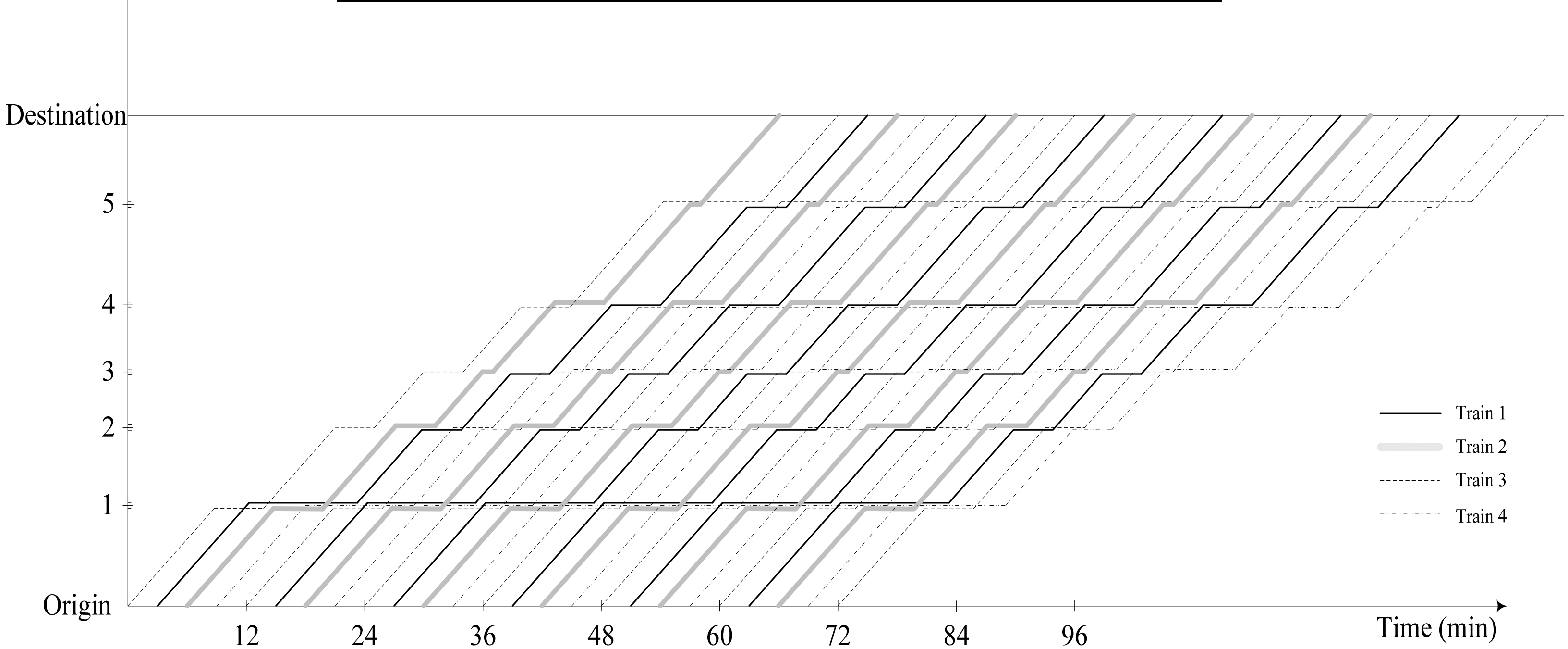


Figure 4.5 Illustrative example #2: 4 train types, 5 stations, and 15 platforms (3 in each station).

Our work regarding the southbound timetable is summarized in Figure 4.6. The top of the figure shows a typical portion of the actual timetable (departure times only). Notice that the timetable has a cycle of 60 minutes with four train types per cycle—two types that stop at all intermediate stations and two types that stop at two intermediate stations. The input parameters in our model are shown on the left of Figure 4.6. The number of platforms at Taoyuan and Taichung stations is known to be 1 and 2 respectively. Based on this information and the structure of the actual timetable, we assume there are 2, 1, 1, 2, 1, and 1 platforms at the intermediate stations respectively. The traveling times between pairs of stations $trav_s$ and minimum dwell times in the intermediate stations $dMin_{ts}$ are inferred from the actual timetable. We assume that w_{ts} = 0 (1) when $dMin_{ts}$ is zero (nonzero). The value of parameter $dMax_t$ is determined by intuition. An optimal solution for this problem instance is displayed in the form of a time-space diagram in the bottom right of Figure 4.6. This solution was obtained in two seconds. Notice that the optimal value of *Interval* (= 19) is greater than *maxLB* (=18) but is much less than the current cycle length of 60. Thus, it appears that the existing track infrastructure in this system can feasibly handle at least three times as many trains as it currently accommodates while maintaining the existing 50-50 mix between “local” and “express” trains.

Portion of actual timetable (southbound)

Train	Taipei	Banqiao	Taoyuan	Hsinchu	Taichung	Chiayi	Tainan	Zuoying
135	10:30	10:38	-	-	11:22	-	-	12:06
637	10:36	10:44	10:57	11:10	11:38	12:02	12:21	12:36
139	10:54	11:02	-	-	11:46	-	-	12:30
641	11:00	11:08	11:21	11:33	12:01	12:26	12:45	13:00
143	11:30	11:38	-	-	12:22	-	-	13:06
645	11:36	11:44	11:57	12:10	12:38	13:02	13:21	13:36

Input parameters

T	S	P
4	6	8

$dmin_{ts}$

Station	Train 1	Train 2	Train 3	Train 4
Banqiao	4	4	4	4
Taoyuan	0	4.5	0	4.5
Hsinchu	0	4.5	0	4.5
Taichung	8	8	8	8
Chiayi	0	7.5	0	7.5
Tainan	0	7.5	0	7.5
$dMax_t$	20	48	20	48

$hTrack_s = 3$ for all s | $hPlatform_p = 1$ for all p

Stations with two platforms:
Banqiao, Taichung

$a_1 = 1$ $a_2 = 0.0001$
$maxLB = 18$

	Train 1	Train 2	Train 3	Train 4
Taipei	10	7	0	17
	Actual Dwell Time			
Banqiao	4	4	4	4
Taoyuan	0	8	0	9
Hsinchu	0	9	0	8
Taichung	8	8	8	9
Chiayi	0	9	0	8
Tainan	0	8	0	8

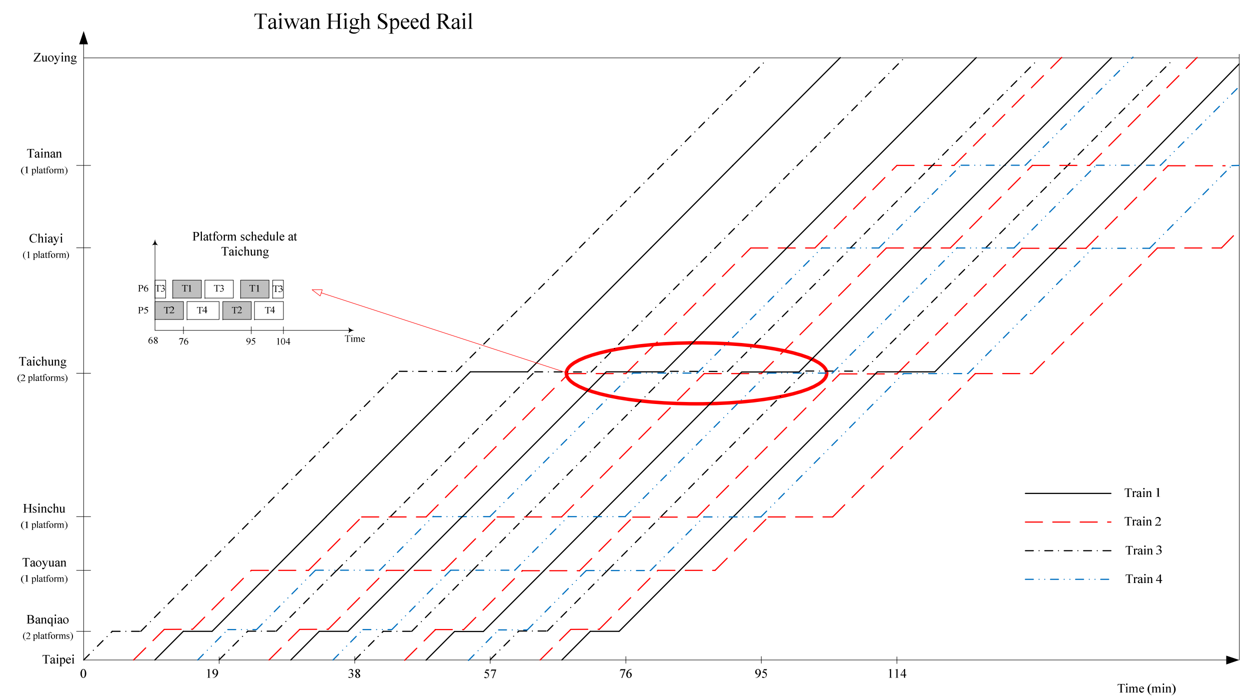


Figure 4.6 Optimal time-space diagram for the Taiwanese high speed railway example.

4.5.4. Fourth illustrative example: Japanese high-speed railway system

The fourth illustrative example is taken from the Japanese high-speed railway system. This system consists of several main lines, the most extensive of which is the Tokaido and Sanyo Shinkansen that runs 1436 km between Kagoshima and Tokyo in western Japan. Three main categories of trains—super express, limited express, and express—run on this line. Trains in these categories stop at roughly 25%, 50%, and 100% of the intermediate stations they encounter respectively. Trains in the same category do not pass each other. Super express trains may pass limited express and express trains, and limited express trains may pass express trains. No other passing combinations are allowed. The five subcategories of trains are the Nozomi super express (N), Mizuho super express (M), Sakura limited express (S), Hikari limited express (H), and Kodama express (K). The current timetable for this line is available on the internet (Japanese high speed rail website, 2012).

The work on this problem considers the eastbound portion of the line that runs from Kagoshima (the origin) to Tokyo (the destination) and has 35 intermediate stations. The midday portion of the timetable strongly resembles a cyclic timetable with a one hour period and 11 train types per cycle. We focus on the final 22 trains in the top row in page 2 of the timetable, i.e. the portion of the timetable beginning with train "K738" and ending with train "H468." This portion of the timetable is almost perfectly cyclic; the final 11 trains are virtually identical to the first 11 trains except they are scheduled 60 minutes later. Each cycle contains 11 train types: two N trains traveling from station 2-36; one N train traveling from station 9-36; one N train traveling from station 20-36; one M train traveling from station 0-20; one S train traveling from station 0-20; two H trains traveling from station 15-36 and 20-36 respectively; and three K trains traveling from station 2-15, 20-36, and 24-36 respectively. Thus, the actual timetable fits within the modeling framework of this study. The work on this problem is summarized in Figure 4.7. Note that *oStationt* $\neq$ 0 or *dStationt* $\neq$ S+1 for all t. We assume there is one platform at every intermediate station. The traveling times between pairs of stations *travs* and minimum dwell times in the intermediate stations *dMints* are inferred from the actual timetable. The value of parameter $dMax_t$ is 19-21 minutes higher than the sum of $dMint_s$ for each t. The standard MILP model is supplemented with additional constraints of type (4-31) and (4-32) that ensure that the two N trains traveling from station 2-36—train types 1 and 10—maintain at least 20 minutes of headway on all portions of the main line.

Figure 4.7 shows an optimal time-space diagram for this problem instance when a_1 = 1, a_2 = 0.0001, and *travs* = 0 for all s. The optimal objective value is 59.0848 (*Interval* = 59 and total train dwell time = 848 minutes). The parameter $trav_s$ is zeroed out so that four intervals worth of trains can be displayed on a single page. Note that the optimal time-space diagram (timetable) for the original problem instance can be obtained by changing the vertical lines to a diagonal orientation (via a simple transposition in which an appropriate sum of $trav_s$ parameters is added to each arrival and departure time in the timetable). This solution was obtained in 10,166 seconds. Notice that the optimal value of *Interval* (= 59) is slightly less than the actual cycle length of 60. Thus, it appears that the existing track infrastructure in this system is being pushed to its limits under the assumed values for $dMint_s$, $hTrack_s$, and $hPlatform_p$. A second experiment considered the same instance with a_1 = 1 and a_2 = 0 and identified the optimal *Interval* (=59) in only 6415 seconds with total train dwell time = 912 minutes. This result indicates that the optimal *Interval* may be found more quickly when the second objective is disregarded.

Here, we fix *Interval* = 60 = *minUB* = *maxLB* and we only pursue objective 2. Our solution was obtained in 5921 seconds. In this case, the minimum total train dwell time is 828 minutes, i.e. 20 minutes less than when *Interval* = 59. This result demonstrates that the two objectives are conflicting and it is possible to obtain substantial reduction with respect to objective 2 at the cost of little increase in objective 1. The runtime result indicates that the problem becomes easier to solve when extra "breathing room" in the form of a fixed, longer cycle length is incorporated into the timetable.

4.6. Additional experiments: setup, results, and discussion

We now perform several additional experiments to confirm the effectiveness of the second model presented in Section 4.3. These experiments consider hundreds of problem instances, most solved to optimality. In all experiments, CPLEX is given a 40 minute time limit for solving the overall math program presented in (4-24) – (4-45). This time is in addition to the time used by CPLEX to solve preliminary machine scheduling problems in each station for calculating maximum lower bound (*maxLB*) (Table 4.3) and any other time needed to set up the problem.

The problem instances are defined by the values of the primary input parameters *S*, *P*, s_p, p_s, *T*, $oStation_t$, $dStation_t$, $trav_s$, w_{ts}, $dMin_{ts}$, $dMax_t$, $hTrack_s$, $hPlatform_p$, and a_k. Table 4.6 shows the parameter values that are considered in this section. The brackets "[*a*,*b*]" indicate the set of integers in the closed interval from *a* to *b*.

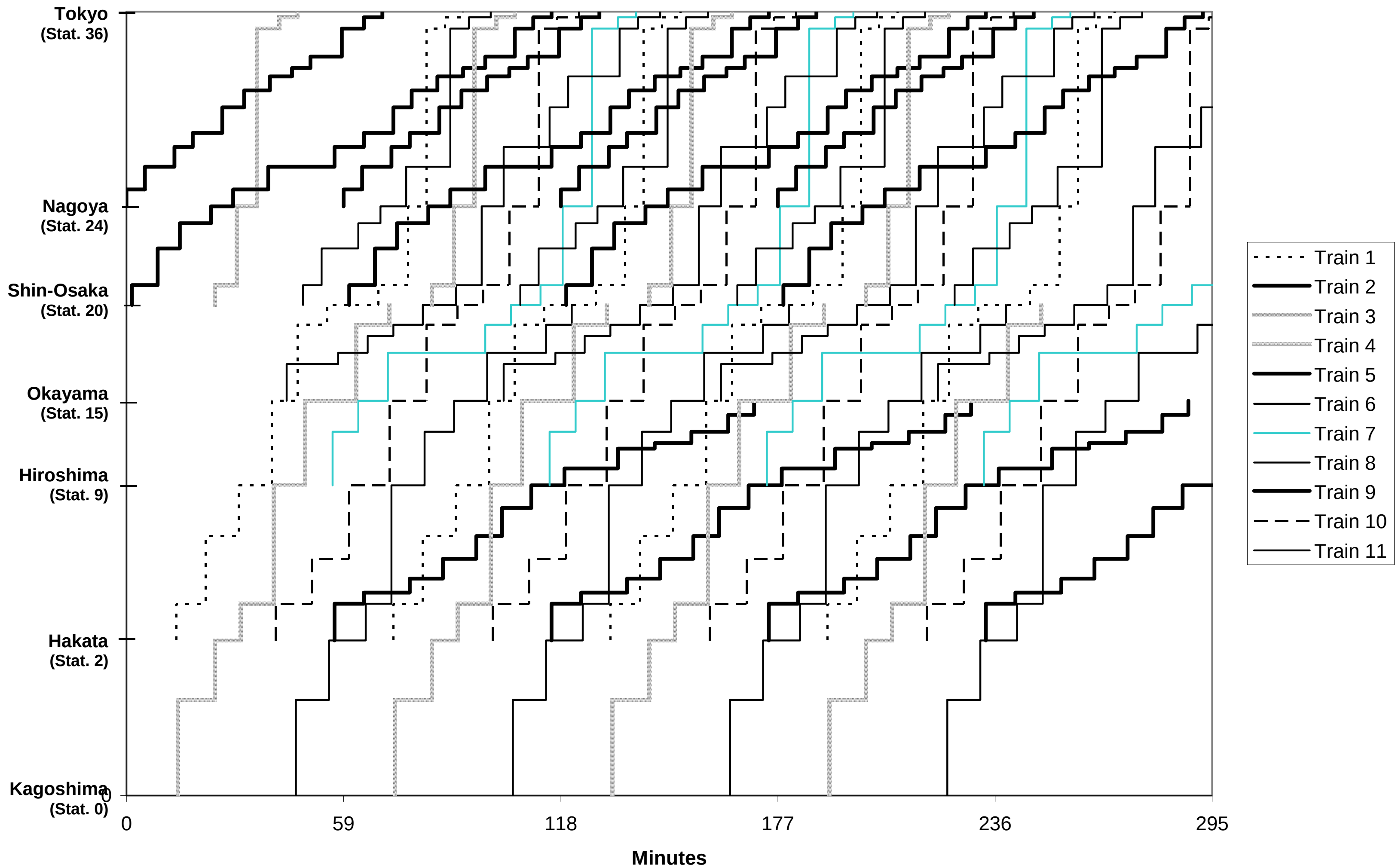


Figure 4.7 Transposed version (*trav*$_s$ = 0 for all *s*) of optimal time-space diagram for the Tokaido and Sanyo Shinkansen—the bullet train line that runs 1436 km from Kagoshima to Tokyo, Japan (S = 35, P = 35, T = 11, *hTrack*$_s$ = 3, *hPlatform*$_p$ = 0.5, a_1 = 1, a_2 = .0001. *Interval* = 59, total dwell = 848.0.)

Table 4.6 Parameter values considered in Section 4.6

Parameter	Value
T	2, 3, 4, 5, 6
S	[1, 33]
P	$S + X$ where $X \sim$ B(S, 0.2) (B = binomial distribution)
$\lvert p_s \rvert$	1, 2
$oStation_t$	0
$dStation_t$	S+1
w_{ts}	Usually = 1 if $\lvert p_s \rvert = 2$; usually = Bernoulli(0.5) if $\lvert p_s \rvert = 1$
$dMin_{ts}$	= 1, 2, 3, 4, 5, or 6 if $w_{ts} = 1$; = 0 if $w_{ts} = 0$
$dMax_t$	$1.5 \times \sum dMin_{ts} + [0, 9]$
$hTrack_s$	1, 2, 3, 4 (held constant on entire main line in each instance)
$hPlatform_p$	0.5
(a_1, a_2)	(1, 0.0001)
$trav_s$	5, 6, 7, 8, 9, 10

Unless otherwise noted, the following procedures are used to determine the parameter values in all experiments in Section 4.6. First, T and S are fixed. Then we perform a Bernoulli trial in each station—there are two (one) platforms in station s with probability 0.2 (0.8). Thus, $P = S + X$ where $X \sim$ B(S, 0.2), a binomial distribution with S independent trials where the probability of success in each trial is 0.2. Then $oStation_t = 0$ and $dStation_t = S$+1 for all t. Then w_{ts} is initially generated so that, for each train type t, the probability that it stops at a station s with 1 (2) platform(s) is 0.5 (1). Each station s without a train stop then "steals" a stop from the closest station with at least two stops. This ensures that each station has at least one train type stopping in it. Then $dMin_{ts}$ for each train stop (i.e. instance where $w_{ts} = 1$) is initially set to a random integer from 1 to 6 inclusive; $dMin_{ts}$ is set to 0 when $w_{ts} = 0$. Next, the non-zero $dMin_{ts}$ values are swapped among themselves so that $(\Sigma dMin_{ts})/\lvert p_s \rvert$ is roughly the same for each station s, i.e. so that the total minimum train dwell time per platform in each station is roughly equal. This makes for a challenging problem instance in which no single station is the obvious bottleneck. Then $dMax_t$ is calculated based on $dMin_{ts}$ as shown in Table 4.6. Finally,

$hTrack_s$, $hPlatform_p$, a_1, and a_2 are set to the values shown in Table 4.6. Note that the parameter $trav_s$ plays only a trivial role in defining a problem instance. Thus, T and S are decided by human judgment and the other parameter values are usually determined automatically.

Note that every combination of parameter values satisfying the above criteria gives rise to a problem instance with a non-empty feasible region. Also, both components of the objective function are minimized and both have zero as a lower bound. Thus, every problem instance has an optimal solution. In the figures showing the experimental results, a solid dot denotes an optimal solution and a hollow dot denotes the best feasible solution that was found when the time limit was reached.

Table 4.7 shows the results from the first set of experiments in which 55 problem instances are considered. The main parameters defining each instance are on the left and the main aspects of the solution are given on the right. The results show that 53 (55) out of 55 problems are solved to optimality within the 40-minute time limit when both objectives are (only the first objective is) considered. Also, the optimal value of *Interval* is usually not equal to the lower bound *maxLB*. Not surprisingly, the runtime generally increases when any one of the parameters S, P, or T increases in isolation. The second last column in Table 4.7 shows the runtime required to find the optimal *Interval* for each problem instance when $a_1 = 1$ and $a_2 = 0$. The results in this column show there is a reduction (an increase) in runtime for 42 (12) problem instances when the second objective is disregarded. Furthermore, the final column shows that the average reduction in runtime is more substantial than the average increase. These results indicate that the problem becomes significantly easier to solve when the second objective is disregarded.

Table 4.7 Results for 55 problem instances ($hTrack_s = 1$)

Input Parameters					Solution					
						Two Objectives			First Objective Only	
Instance #	*S*	*P*	*T*	*maxLB*	*Interval*	Total Dwell	Time (sec)	Status	Time (sec)	Change
1	5	5	3	12	14	44.5	3	Optimal	2	-1
2	5	5	3	12	14	49	4	Optimal	3	-1
3	5	6	4	11	12	53.5	5	Optimal	7	+2
4	6	8	4	13	13	75.5	16	Optimal	4	-12
5	6	8	4	13.5	13.5	77.5	27	Optimal	15	-12
6	7	7	4	11	12.5	58	6	Optimal	4	-2
7	7	7	4	11	15	56.5	5	Optimal	5	same
8	8	9	3	12	12	72	5	Optimal	3	-2
9	8	9	4	11.5	14	83	14	Optimal	9	-5
10	8	11	4	13	13.5	104	29	Optimal	95	+66
11	9	9	3	13	13	92	5	Optimal	3	-2
12	9	10	3	12	14.5	95.5	6	Optimal	4	-2
13	9	10	4	13	16	105.5	14	Optimal	13	-1
14	10	10	4	13	14.5	100.5	11	Optimal	5	-6
15	10	11	3	12	16	103	7	Optimal	5	-2
16	10	13	3	13	13	105.5	7	Optimal	6	-1
17	10	13	3	12.5	13.5	98.5	12	Optimal	18	+6
18	11	11	4	12	15	117.5	14	Optimal	9	-5
19	11	11	4	13	15.5	112.5	13	Optimal	9	-4
20	12	12	4	13	16	135.5	17	Optimal	10	-7
21	12	12	4	12	14	132.5	20	Optimal	9	-11
22	12	13	3	13	14.5	129	8	Optimal	6	-2
23	12	13	4	13	14	134	19	Optimal	11	-8
24	12	13	4	13	17.5	150.5	23	Optimal	24	+1
25	12	14	3	13	14.5	138	8	Optimal	5	-3
26	12	14	4	13	14	145	38	Optimal	56	+18
27	12	14	4	12	16	137	55	Optimal	66	+11
28	13	13	3	12	16	137	9	Optimal	5	-4
29	13	13	4	13	15.5	129	31	Optimal	15	-16
30	13	13	4	13.5	15.5	137	23	Optimal	17	-6
31	13	13	4	13	17	142.5	20	Optimal	10	-10
32	14	15	3	12	12	127.5	8	Optimal	5	-3
33	14	16	3	13	13.5	141	13	Optimal	9	-4
34	15	16	4	14	15.5	153	25	Optimal	18	-7
35	15	18	3	12	13	159	13	Optimal	10	-3
36	16	18	3	12	14.5	166	16	Optimal	11	-5
37	16	18	3	13	16.5	177.5	13	Optimal	9	-4
38	17	18	4	13	16.5	186	736	Optimal	433	-303
39	17	19	4	12	14.5	182.5	363	Optimal	282	-81
40	17	20	3	13	15	188	18	Optimal	15	-3
41	18	19	4	12.5	15.5	204	72	Optimal	70	-2
42	18	20	3	13	14.5	187.5	58	Optimal	94	+36
43	18	21	4	13	14.5	210	510	Optimal	735	+225
44	19	20	3	13	16	190.5	17	Optimal	19	+2
45	19	21	3	13	14	203	18	Optimal	14	-4
46	19	22	3	13	16	200	29	Optimal	39	+10
47	20	21	3	12	14	196	24	Optimal	34	+10
48	20	22	4	13	15.5	230	1655	Optimal	173	-1482
49	20	23	4	13	14.5	227	2440	Feasible	374	-2066
50	21	22	4	13	16.5	236	971	Optimal	365	-606
51	22	23	4	13	16	230.5	1484	Optimal	923	-561
52	22	24	4	13	16.5	257	259	Optimal	206	-53
53	23	24	4	13.5	17	243	154	Optimal	218	+64
54	24	28	3	13	13.5	263	376	Optimal	108	-268
55	26	29	4	13.5	17	321	2410	Feasible	754	-1656

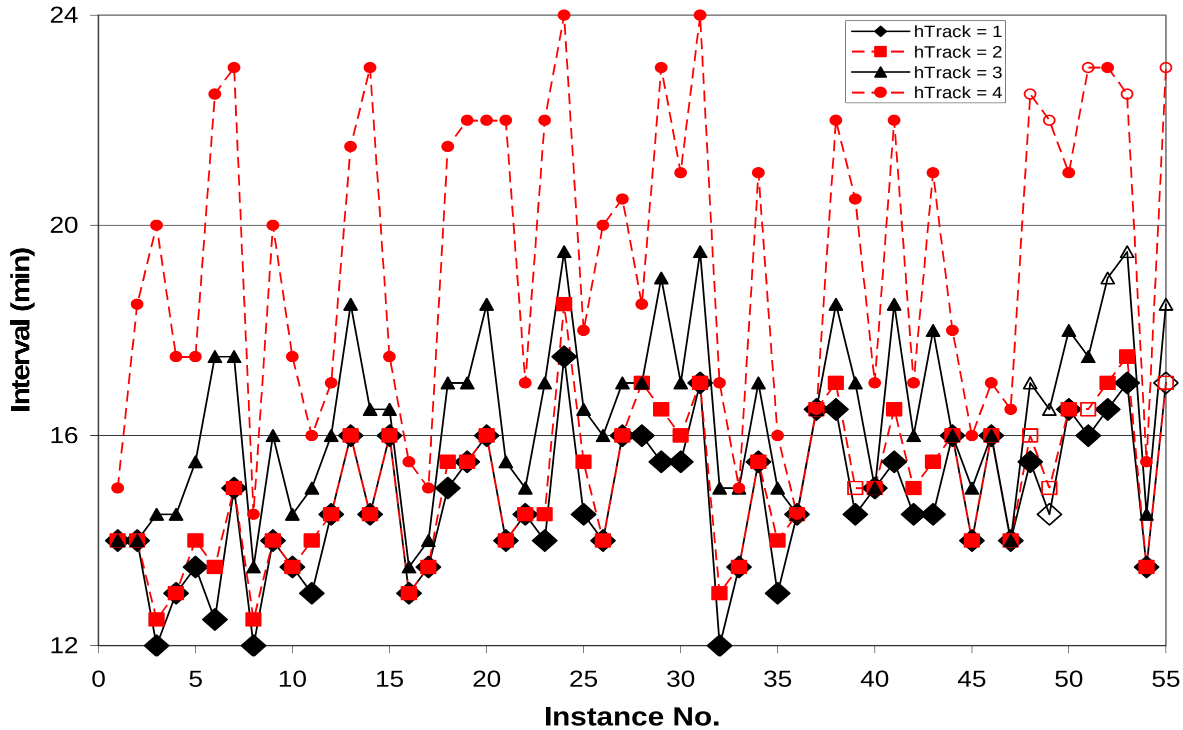


Figure 4.8 Effect of *hTrack*$_s$ on *Interval* for the 55 problem instances considered in Table 4.8.

Figure 4.8 displays the results of a second set of experiments that show the impact of *hTrack*$_s$ on the optimal value of *Interval*. Here we consider four different values of *hTrack*$_s$—1, 2, 3, and 4—in each of the 55 problem instances from Table 4.7. Thus, we consider 220 problem instances total, and the same value of *hTrack*$_s$ is applied to the entire main line in each instance. The results for the cases with *hTrack*$_s$ = 1 are copied from Table 4.7. Not surprisingly, the results show that the optimal value of *Interval* increases monotonically as *hTrack*$_s$ increases. However, the amount of increase in *Interval* depends on the problem instance. For example, there are problem instances in which (i) *Interval* increases steadily as *hTrack*$_s$ increases (instance 11); (ii) *Interval* increases only when *hTrack*$_s$ changes from 2 to 3 (instance 33); and (iii) *Interval* is the same for *hTrack*$_s$ = 1, 2, 3, and 4 (instances 36 and 37).

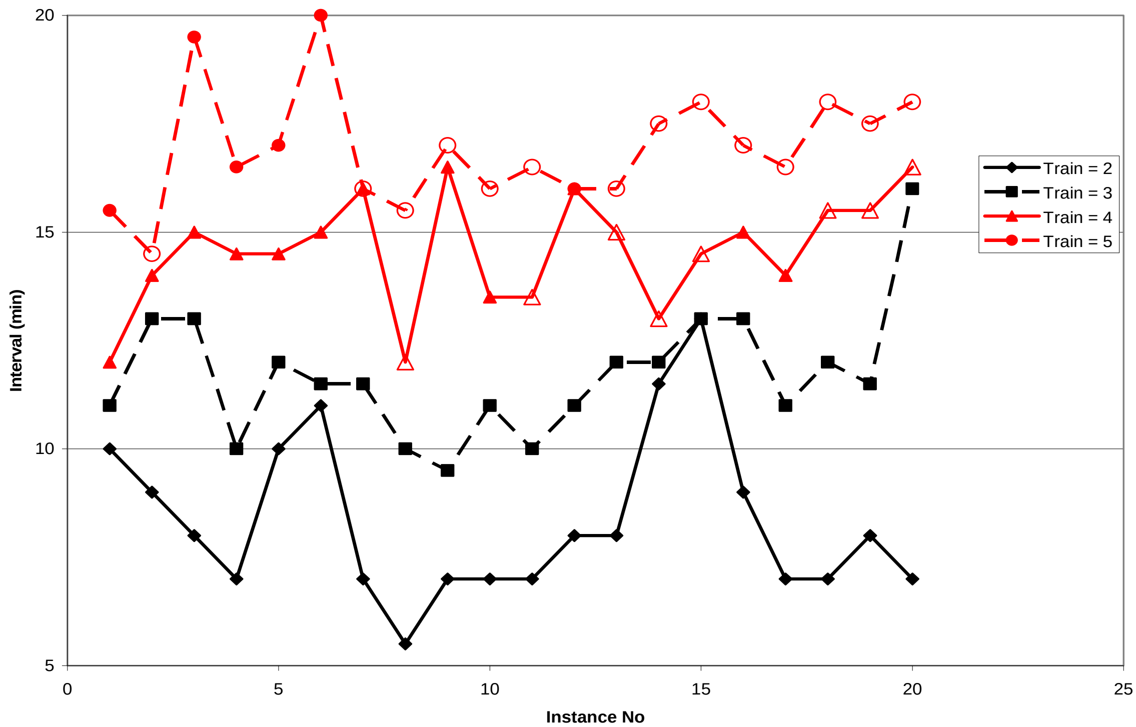


Figure 4.9 Impact of adding extra train types to 20 base problem instances ($hTrack_s$ = 1).

The next two experiments investigate the effect of the number of train types on the objective function. Figure 4.9 displays the results of a third set of experiments that show how the optimal value of *Interval* changes when extra train types are added to a problem instance. Here, we consider 20 base instances and four values of T—2, 3, 4, and 5—for each base instance. Thus, we consider 80 instances total. Each instance has between 10 and 19 stations. Each base instance has two train types (T = 2) and extra train types are added to it one-by-one. That is, the input data for each instance with n train types (n = 3, 4, 5) and is identical to the input data for the corresponding instance with n-1 train types except that one additional train type—defined by w_{ts}, $dMin_{ts}$, and $dMax_t$—is considered. The results show that the optimal value of *Interval* increases monotonically as new train types are added to an existing problem instance. However, the amount of increase in

Interval depends on the base instance and on the exact specifications of the added train type. Indeed, sometimes the optimum *Interval* is unchanged when a new train type is added (e.g. base instances 12 and 15) and sometimes the optimum *Interval* changes drastically (e.g. base instances 3 and 6) when a new train type is added. Figure 4.10 shows the total runtime for the instances considered in Figure 4.9. Here, we see a dramatic increase in runtime as train types are added, which agrees with Table 4.7 and completes our conclusion.

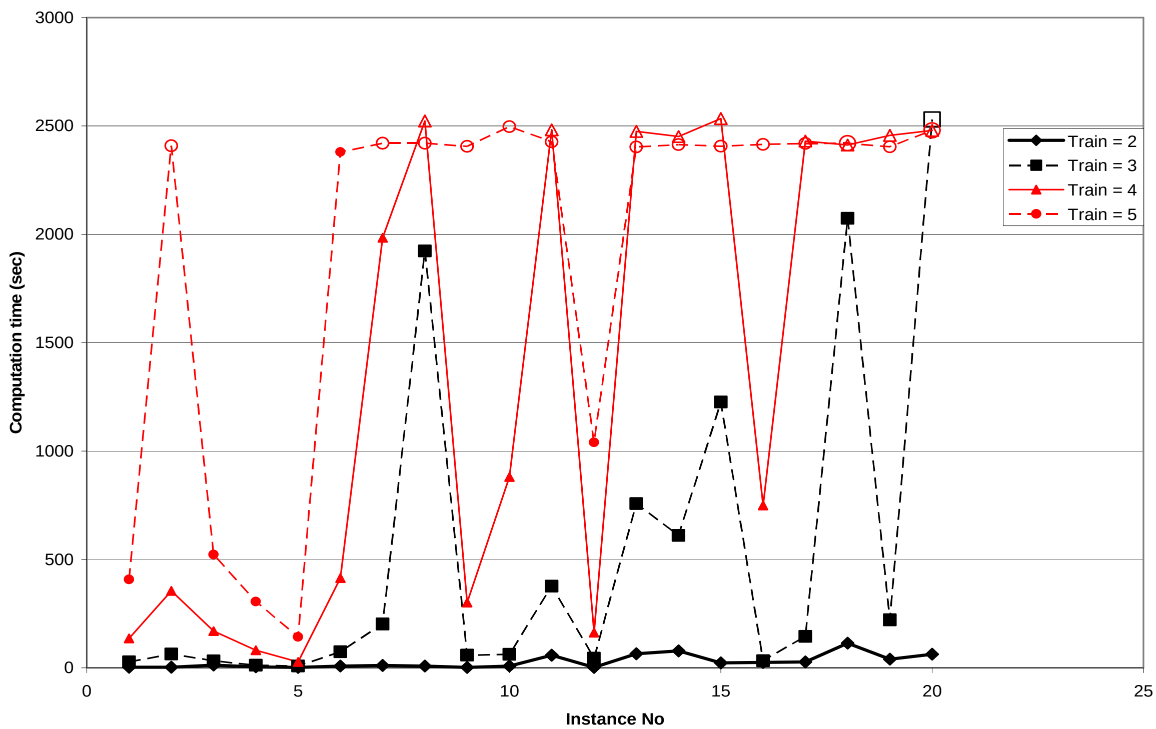


Figure 4.10 Computation times for instances considered in Figure 4.9

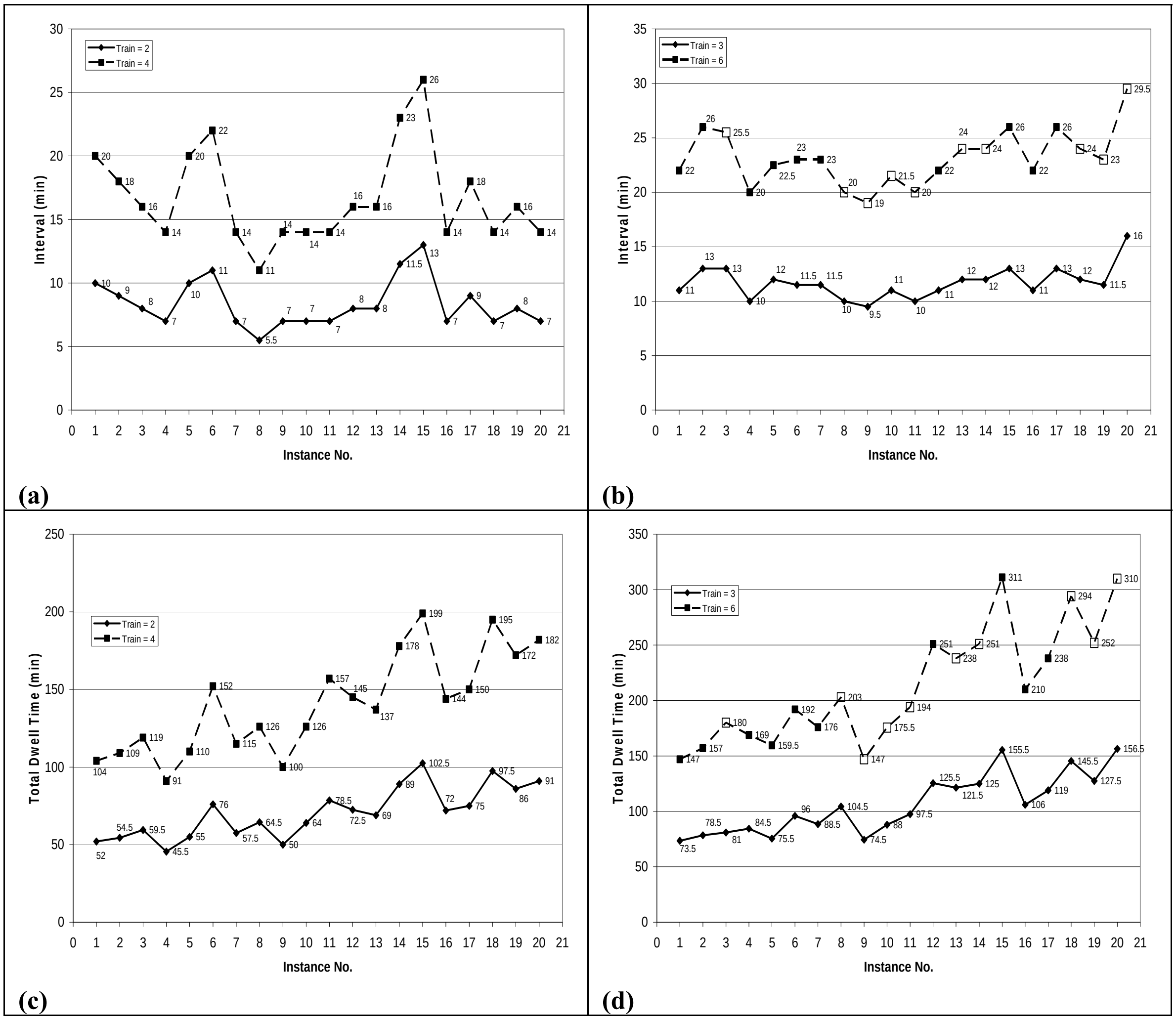


Figure 4.11 Impact of doubling the number of train types via cloning for 20 base problem instances with 2 train types (a, c) and 20 base problem instances with 3 train types (b, d).

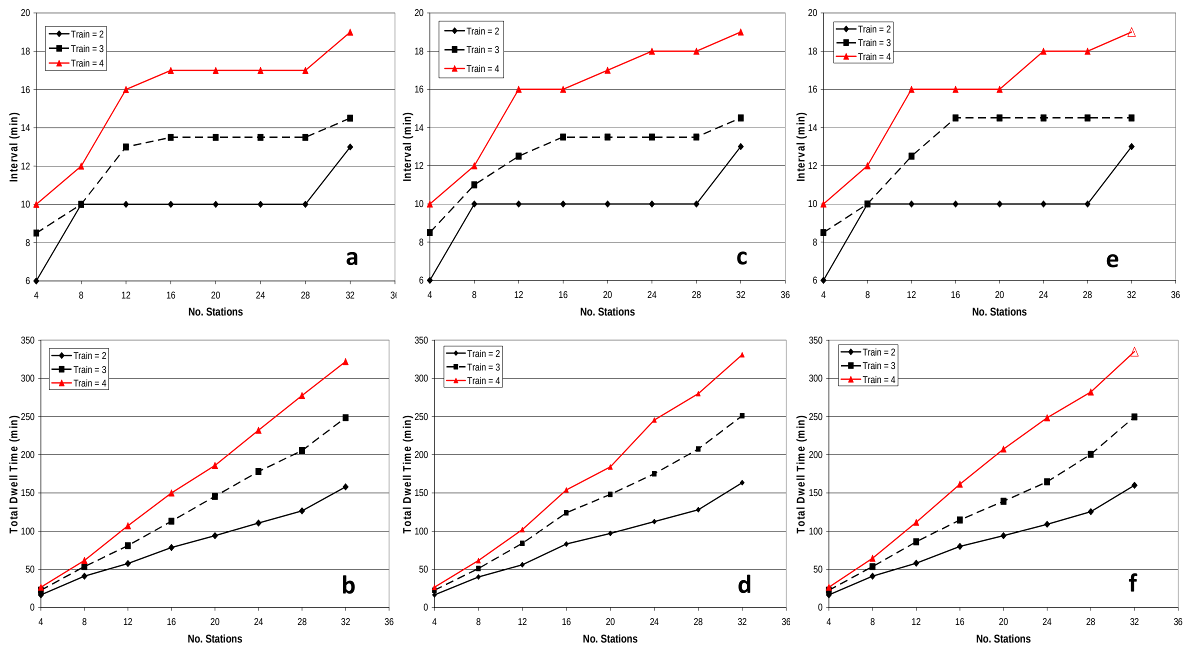
Figure 4.11 displays the results of a fourth set of experiments that show how the two parts of the objective value—*Interval* and total train dwell time—are affected when we double the number of train types via cloning. Here, we consider 40 base instances—20 with $T = 2$ and 20 with $T = 3$. These base instances are identical to the corresponding instances with $T = 2$ and $T = 3$ in Figure 4.9 (instances 16 and 17 appear in reverse order). Each base instance forms the basis of a larger "twin" instance whose train types consist of two exact copies of each train type in the base instance. Thus, we consider 80 instances total, each having between 10 and 19 intermediate stations. The results show that the optimal value of *Interval* and the associated total train dwell time for the twin instance are usually exactly double that of the corresponding base instance. However, there are several exceptions to this general trend. For example, in instance 15 with 2/4 train types, *Interval* (total train dwell time) for the twin instance is double (less than double) that of the base instance. Also, in instance 5 with 3/6 train types, *Interval* (total train dwell time) for the twin instance is less than double (more than double) that of the base instance. Finally, in instances 3 and 10 with 3/6 train types, both *Interval* and total train dwell time for the twin instance are less than double that of the base instance. Thus, it is sometimes possible to increase line capacity by doubling the number of train types dispatched per cycle via cloning.

Figure 4.12 displays the results and impact of adding extra stations to the end of an existing line on *Interval* and total dwell time which is the fifth set of experiments. The three base instances have $T = 2$, 3, and 4 with $S = 4$. The results for the base instances are copied into the left-most portion of parts (a), (b), and (c) in the figure. These base instances are extended by adding four stations to the line at a time in three different ways.

In part (a), two intermediate stations are added to the beginning of the line (i.e. prior to station 1) and two stations are added to the end of the line (i.e. after to station S) in sequential fashion until the total number of stations is 32. In part (b) four stations are added to the beginning of the line in sequential fashion until S = 32. In part (c) four stations are added to the end of the line in sequential fashion until S = 32. Thus, Figure 4.12 summarizes the results from a total of (3)(1 + 7*3) = 66 problem instances. Each station has one platform in all instances (i.e. $P = S$). The instances are constructed so there is a maximum sharing of input data. That is, the input data for every instance with n train types (n = 3, 4) is identical to the input data for the corresponding instance with n-1 train types except that one additional train type—defined by w_{ts}, $dMin_{ts}$, and $dMax_t$—is considered. Also, the input data for every instance with n stations ($n \geq 8$) is identical to the input data for the corresponding instance with n-4 stations except that four additional stations—defined by $hTrack_s$, $hPlatform_p$, $trav_s$, w_{ts}, and $dMin_{ts}$—are considered. Finally, the input data for the stations in parts (b) and (c) are used to form the input data for the stations in part (a). Not surprisingly, the results show that both *Interval* and total dwell time generally increase as stations are added to the ends of an existing line regardless of whether these stations are added at the beginning, at the end, or both at the beginning and at the end of the line.

The sixth set of experiments investigates the effect of number of platforms on *Interval* (rail line capacity) by adding extra platform to the station of an existing line one-by-one. Figure 4.13 displays the result of this investigation. The result for the base instance—in which T = 3, S = 15, and P = 15—is displayed on the left of. Here, we explore two methods for inserting additional platforms into the base instance. In both methods, platforms are added to the instance one-by-one until all stations have two platforms. The bold line shows the result when platforms are added to the stations according to their position along the line, i.e. in sequential fashion beginning with station 1 and ending with station 15. The dashed line shows the result when platforms are first added to the stations s with the highest $\sum_{t=1}^{T} dMin_{ts}$. That is, platforms are first added to the stations that "need" them the most. Thus, Figure 4.13 shows the results for a total of (1 + 14×2 + 1) = 30 problem instances. Each dot in the figure is labeled with the number of the station in which the extra platform is installed. The results indicate that the strategic placement of only one or two extra platforms in particular stations along a rail line may have a greater impact on capacity than the comprehensive placement of many extra platforms along a large portion on the line. Also, the increase in line capacity obtained by adding one platform at a particular station (e.g. station 9 or 15) depends on the number of platforms already installed at other stations.

Figure 4.14 displays the results from a final set of experiments that show the impact on *Interval* of increasing $dMax_t$ above $\sum_{s=oStation_t+1}^{dStation_t-1} dMin_{ts}$ for six base instances. The main input parameters for each base instance—in the format (S_P_T)—are shown in the upper right portion of the figure. Each base instance is constructed so that $dMax_t = \sum_{s=oStation_t+1}^{dStation_t-1} dMin_{ts}$ for all t. The results for the base instances are shown on the left. Other instances are obtained increasing the $dMax_t$ for all t in the base instance by the amount shown on the x-axis while holding all other parameter values (including $dMin_{ts}$) constant. Thus, the figure shows the results for a total of [6×10 = 60] problem instances. The results show that the optimal value *Interval* is a decreasing and generally convex function of the $dMax_t$. The large decreases in *Interval* when the $dMax_t$ are initially increased above the $\sum dMin_{ts}$ are attributed to the fact that the decision maker has no flexibility when the $dMax_t = \sum dMin_{ts}$. In this case, adding just a little flexibility to a case with no flexibility leads to a substantial improvement in *Interval*. Note that, in all problem instances, no further improvement in *Interval* is achieved when the $dMax_t$ exceed $\sum dMin_{ts}$ by more than ten minutes. In other words, we can often obtain the optimal value of *Interval* without much increase in train dwell times above the minimum required train dwell times.



(a)-(b) 2 stations added at the beginning and 2 at the end

(c)-(d) 4 stations added at the beginning

(e)-(f) 4 stations added at the end

Figure 4.12 Impact of adding 4 stations at a time on *Interval* and total dwell time (*P* = *S*).

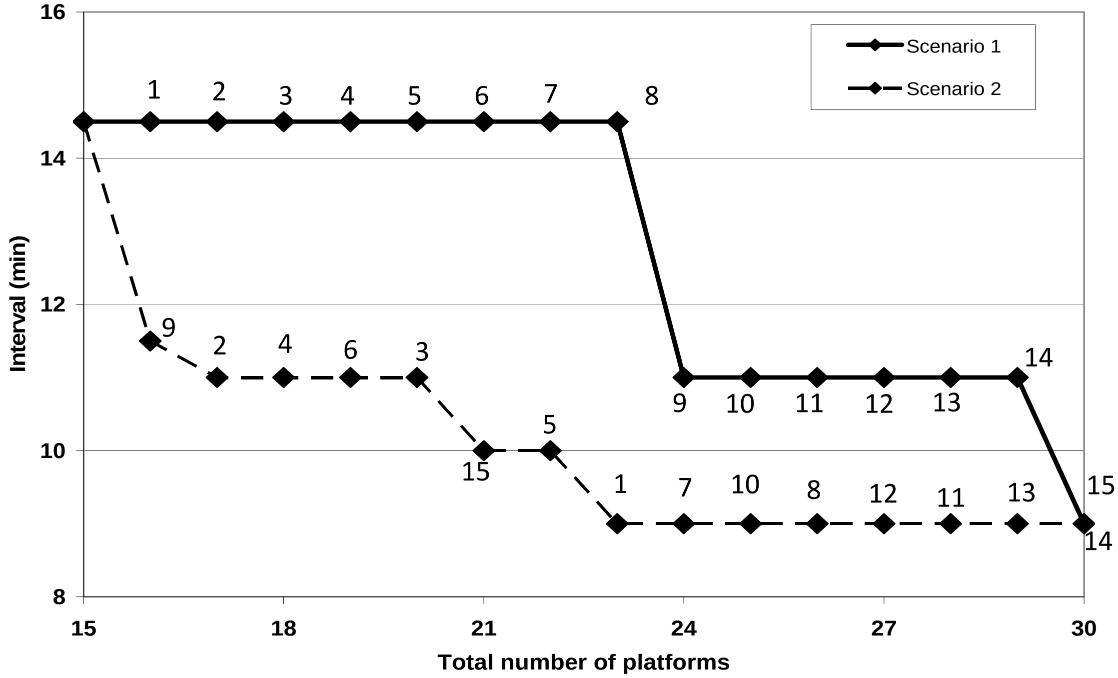


Figure 4.13 Impact on *Interval* of adding platforms to the stations of an existing line (labels indicate stations were an extra platform is installed).

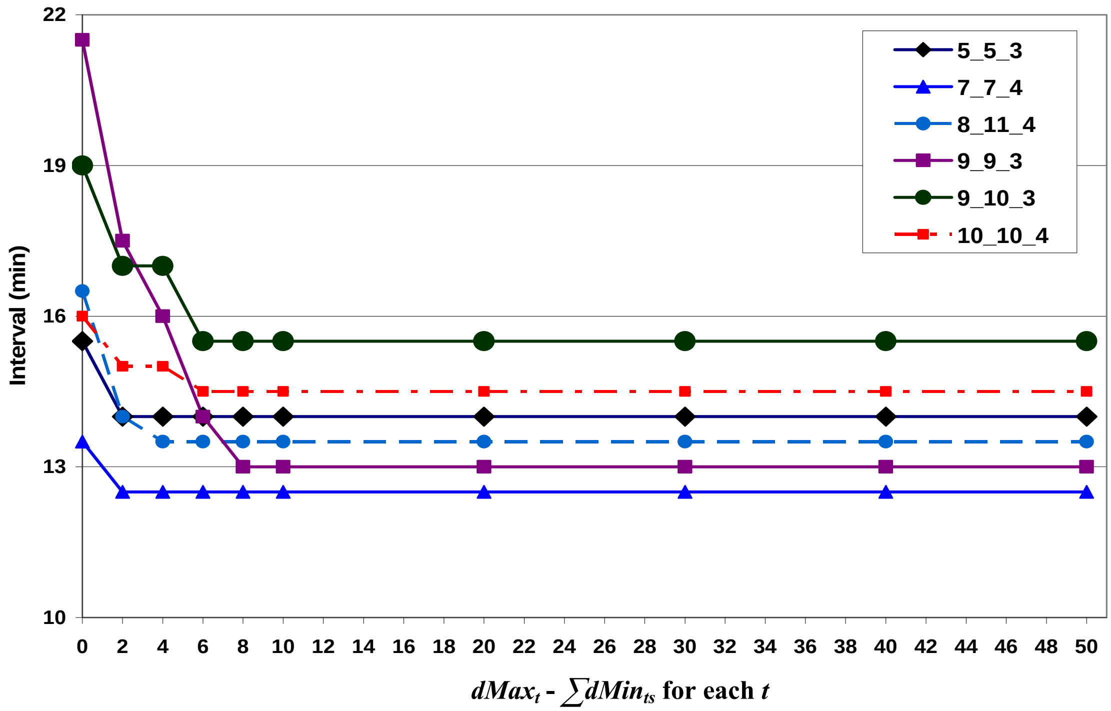


Figure 4.14 Impact on *Interval* of increasing $dMax_t$ above $\sum dMin_{ts}$ for six base instances.

4.7. Conclusion

In this chapter we have presented two mixed integer linear programming models of a cyclic, combined train timetabling and routing. This is the first attempt of integrating cyclic train timetabling and routing by mixed integer linear programs in the literature. These MILP models schedule train arrivals and departures at stations and assigns train types to platforms in the stations so as to minimize the length of the dispatching cycle and/or minimize the total stopping (dwell) time of all train types at all stations combined. The first objective—minimization of the length of the dispatching cycle—directly relates to rail line capacity. The current study generalizes the model presented in Chapter 3 in three ways. First, we consider any number of train types per cycle. Second, we allow stations to have more than one siding. Third, we allow trains to start or end at intermediate stations. Two real-world problems along with hundreds of randomly generated and real-world problem instances have been considered and solved to optimality in a reasonable amount of time using IBM ILOG CPLEX 11.2 and 12.4.

The experimental results yield several managerial insights. First, our ability to solve large problem instances to optimality—including an instance with 11 train types and [33] intermediate stations taken directly from the Japanese Shinkansen bullet train system timetable—demonstrates the effectiveness of the model. Second, problems of this type generally become more difficult when the number of stations, platforms, or train types increases. On the other hand, the problem becomes significantly easier to solve when (i) the second objective—minimizing total train dwell time—is disregarded or (ii) the dispatching cycle is fixed to a value with some "breathing room" and only the second objective is considered. Not surprisingly, the optimal cycle length increases monotonically as (i) the minimum required headway on the main line increases, (ii) extra train types are added to an existing rail line, or (iii) extra stations are added to the ends of an existing line. The optimal cycle length decreases monotonically as (i) extra platforms are added to the stations in an existing rail line or (ii) actual train dwell times are allowed to deviate by a greater amount from their respective minimum required dwell times in the stations where they stop. In addition, it is sometimes possible to increase the line capacity by doubling the number of train types dispatched per cycle via cloning. Finally, the optimal value of *Interval* can often be obtained without sacrificing too much in the form of increased train dwell times above the minimum required train dwell times in the stations.

Chapter 5

Bi-directional cyclic train timetabling and platforming with heterogeneous rolling stock

5.1. Introduction

This chapter presents a new mixed integer linear program for the cyclic train timetabling and platforming for a single-track bi-directional rail line between two major cities known as west and east terminals with heterogeneous rolling stock. This model extends the problem presented in Chapter 4 in four different ways: (i) train types can travel in both directions, (ii) some train types can stops at stations other than those specified by their line plan, (iii) some train types are forbidden to stop at some stations, and (iv) train types have different travel times on the main line and the actual travel time is a decision variable which will be determined by the model. From (ii) and (iii) it can be inferred that the model will determine where a particular train type stops in order to increase line capacity.

Figure 5.1 illustrates the problem setting at hand. Consider a single track, bi-directional rail line with two endpoints and a set of S intermediate stations between them. It is assumed that the stations are labeled from 0 to S+1 from west to east. Further, it is assumed that some stations may have more than one platform allowing more trains to reside in them at any given time. Therefore, there are P platforms such that $P \geq S$. A total of T different train types with their respective starting (*oStation*) and ending (*dStation*) points and station stopping patterns (w_{ts}) are dispatched in cyclic fashion with one train type in each cycle. There are two subsets of trains denoted by T_1 and T_2, respectively, where T_1 is the set of eastbound trains and T_2 is the set of westbound trains $(T_1 \cup T_2 = T)$. The starting point or origin ($oStation_t$) of train type t may be either of the endpoints (i.e. "station 0" or "station S+1") or any intermediate station. The destination ($dStation_t$) of train type t may be the other endpoint (other than starting point, i.e. "station S+1" or "station 0") or any station after $oStation_t$ (if t is an eastbound train) or before $oStation_t$ (if t is a westbound train). Each train type, as specified by its line plan, must stop for minimum time at each station or it might be forbidden to stop at a particular station. Without loss of generality, this minimum time, also known as dwell time, can be zero meaning that the stop has not been defined for the train at the time of defining line plan. Therefore, if it is not forbidden, the model can determine whether the train stops or not, i.e. those trains with zero dwell time can stop at the station in order to facilitate other trains' movements on the main line and/or through stations. This characteristic enhances system flexibility and may shorten cycle length depending upon the degree of the importance of the first objective. The train type, in our definition, may relate to direction, stopping frequency, train-station compatibility, or some other differentiating characteristics between train sets.

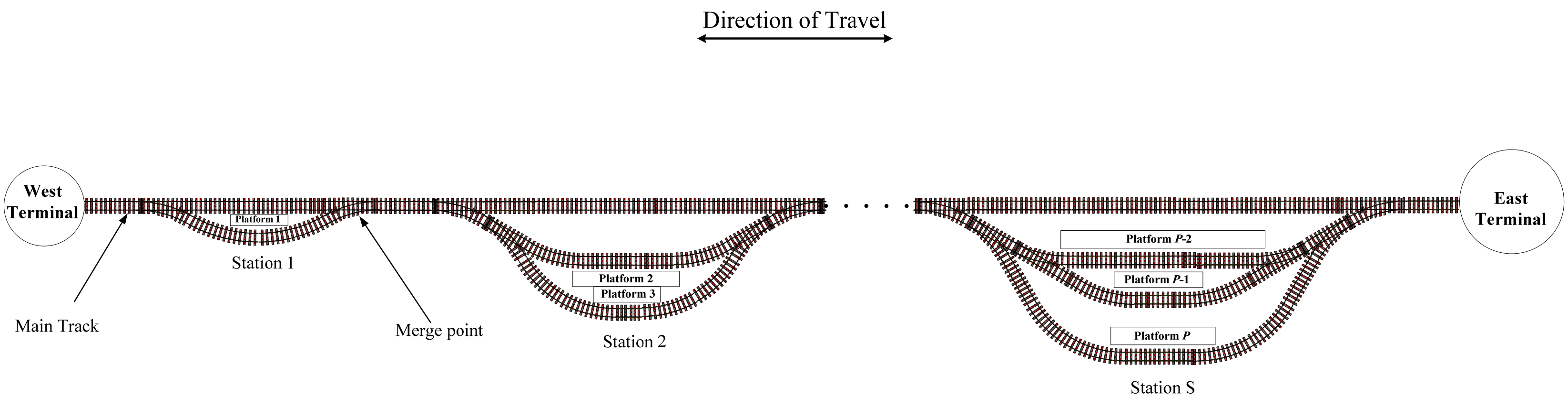


Figure 5.1 Topology of the railway system investigated in this study.

The general feature of the rail line is the same as the problem investigated in Chapter 4 except the bi-directional train movement and heterogeneous rolling stocks, but for the sake of simplicity will be restated here. The single track makes it impossible for one train to pass another if both trains are on the main line. However, there is at least one siding in each station, so it is possible for a train on the main line to pass a train that is stopped at a station. Regarding passing, we assume that a through train's passage of a station is entirely unobstructed by any trains that are stopped at the station. Further, each station siding is sufficiently long so that deceleration and acceleration by a train moving into or out of a station does not interfere with the operations of the through trains that do not stop at the station.

Each train type t is dispatched from its starting point once per cycle and the cycle length, a decision variable, is called *Interval*. The departure (arrival) time from (at) station s is the time when the train reaches the merge point—the point at which the sidings in station s return to the main track just after (before) station s. The cyclic nature of the timetable means that the journeys of all trains of type t are identical except for their departure times from their starting point.

As mentioned earlier in this chapter, in this problem, it is also assumed that each train type's traveling time on the main line between two stations is a decision variable which is bounded below. The total travel time of each train type, referred to as journey time in this paper, is bounded by an upper bound. This upper bound on each train journey allows train to stop at stations for a longer time or move over the main track at slower speed.

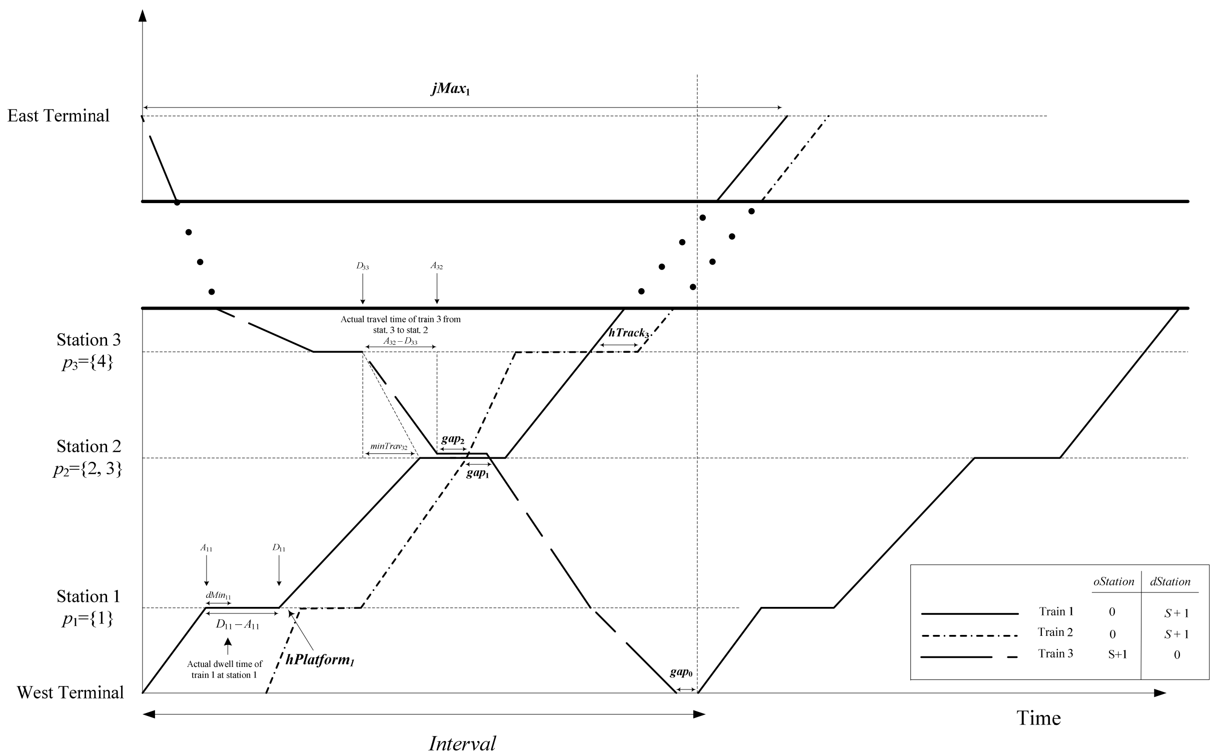


Figure 5.2 Cyclic train timetable depicted as a time-space diagram.

Here, again, the primary objective is to minimize *Interval* subject to five sets of operational and safety constraints which shape this problem. First, there is a minimum dwell time at each station for those train types that are allowed to stop in a station. Some train types are not specified to stop, meaning dwell time is set to zero, but depending on the system situation those train types may stop if doing so lowers the objective function value and/or clear a conflict. The second constraint is related to traveling time on the main line. According to this constraint each portion of the main line or a train type is subject to a speed limit. Third, each train type's total journey time from origin to destination cannot exceed an upper bound. In the fourth constraint—a safety-type constraint—trains on the portion of the main track between station s and s+1 heading the same direction must be separated by a minimum headway of $hTrack_s$ minutes. Similarly, the opposite direction trains must be separated by the minimum safety time which is called gap_s in this dissertation. The last set of constrains are related to safety consideration in stations. According to these safety constraints or regulations every pair of train types that are assigned to the same platform must be separated by a safety time, denoted by $hPlatformSame_p$ for trains going the same direction and $hPlatformDiff_p$ for trains going in opposite direction, respectively. In other words, the departure time of a train type leaving from platform p must be separated by at least $hPlatformSame_p$ ($hPlatformDiff_p$) minutes from the arrival time of the next train for same-direction (opposite-direction) trains. For simplicity, we assume that a train's "departure/arrival time" from/to station occurs at the very end/beginning of its dwell time in the station. The secondary objective is to minimize the total journey time of all train types combined. Figure 5.2 shows a time-space diagram of the type of cyclic train timetable described here in which $|T| = 3$ where $T_1 = \{1,2\}$ and $T_2 = \{3\}$. Each train type is depicted using a different style line (e.g. solid, dashed). Note that the most of the terms introduced in the preceding paragraph—including *Interval*, the length of the dispatching cycle—appear in the figure.

5.2. Brief literature review

As mentioned in this chapter we focus on timetabling optimization (i.e. train scheduling) and train platforming (i.e. track allocation) where different train types have different speeds on the mail line and different minimum dwell times at stations. Some efforts that considered bi-directional track include Ceder (1991), Jovanović and Harker (1991), Higgins et al. (1996), Higgins et al. (1997), Harrod and Schlechte (2013), Caprara et al. (2006), Shafia et al. (2012), and Zhou and Zhong (2007). Recent contribution that just consider train routing/platforming but not timetabling in any form include Zwaneveld et al. (2001), Billionnet (2003), Lusby et al. (2011b), and Caprara et al. (2011). Demagne et al. (2012) model the track assignment problem by online graph coloring and study the computational complexity of the problem. No train scheduling or timetabling problem is considered.

Among all previous works, Bergmann (1975), Heydar et al. (2013), and current study are the only studies that apply mixed integer programming for cyclic train timetabling with the minimization of cycle length as the primary objective. This requires the cycle time to be a decision variable. This is another characteristic that makes their work and the current study different from other researches in the field of train timetabling. This chapter further generalizes the cyclic timetabling problem by considering train movement in both directions on the mail line.

In summary, a close examination of the railway operations literature has yielded many outstanding contributions in the areas of train timetabling, train scheduling, train platforming, and railway capacity analysis. However, the problem investigated in this chapter appears to be the first study to (1) present a MILP model for a cyclic timetabling problem for a single-track bi-directional rail line in which the length of the cycle is a decision variable and objective function; (2) present a model that allows train types to move at different speed; (3) present a model that allows train types stop at stations other than their line plan; and (4) all other aspects of problems considered in Chapter 3 and Chapter 4, respectively.

5.3. Mathematical formulation

We now present an MILP formulation for the problem. The indices, parameters, and decision variables are given in Table 5.1. There are two types of parameters for this model. The primary parameters define a problem instance and can be read from an external file or database. The secondary parameters are derived from the primary ones and are described after presenting mathematical model. Table 5.2 defines the equivalent mathematical expressions for index ranges given in Table 5.1.

Table 5.1 Indices, parameters, and decision variables in mathematical model

Indices:	
s, m	Station index ($0 \le s,\ m \le S+1$; 0 and S+1 represent the west terminal and east terminal)
P	Platform index ($1 \le p \le P$)
D	Train direction (1 = Eastbound or 2 = Westbound)
t, u	Train type index ($1 \le t,\ u \le T$, $t \in T_1$ are eastbound trains and $t \in T_2$ are westbound trains)
I	Interval index; index for different trains of same type (integer; 0 = original interval)
K	Objective function component (k = 1, 2)
Parameters:	
S	Number of intermediate stations (integer, ≥ 1)
P	Number of platforms (integer, $P \ge S$)
s_p	Station in which platform p resides (p = 1 to P)
p_s	Set of platforms in station s (s = 1 to S)
T_d	Number of train types traveling in direction d (d = 1, 2)
T	Number of train types (integer, ≥ 1, $T = T_1 + T_2$)
d_t	Direction for train type t (t = 1 to T)
$oStation_t$	Origin station for train type t (integer, $\ge 0, \le S+1$) (t = 1 to T)
$dStation_t$	Destination station for train type t (integer, $\ge 0, \le S+1$) (t = 1 to T)
$dMin_{ts}$	Minimum dwell time for train type t at station s (real, ≥ 0) ($\forall t, \forall s$: $1 \le t \le T_1$ and $oStation_t < s < dStation_t$ OR $T_1+1 \le t \le T$ and $dStation_t < s < oStation_t$)
$forbid_{ts} = \begin{cases} 1 \\ 0 \end{cases}$	If train type t is forbidden from stopping in station s Otherwise (binary) (same index ranges as for $dMin_{ts}$)
$jMax_t$	Maximum allowed journey time for train type t from its starting point to its ending point (minutes) (real, > 0) (t = 1 to T)
$travMin_{ts}$	Minimum possible travel time for train type t on main line between station s and s + 1 (real, > 0) ($\forall t, \forall s$: $1 \le t \le T_1$ and $oStation_t < s < dStation_t$ OR $T_1+1 \le t \le T$ and $dStation_t < s < oStation_t$)
$hPlatformSame_p$	Headway on platform p for trains traveling in same direction (real, ≥ 0) (p = 1 to P)
$hPlatformDiff_p$	Headway on platform p for trains traveling in opposite directions (real, ≥ 0) (p = 1 to P)
$hTrack_s$	Headway between trains traveling in same direction on portion of main line between station s and s+1 (real, > 0) (s = 0 to S)
gap_s	Minimum time separation at any point along portion of main line between station s and s+1 for trains traveling in opposite directions (real, > 0) (s = 0 to S)
a_k	Weight for objective function component k (real, ≥ 0) (k = 1, 2)
minUB	Minimum upper bound on optimal value of *Interval* (real, > 0)
$mainLB_s$	$= \max_s$ (s=1 to S) $\left(\left(A_{tfast,s} - D_{tslow,s-1}\right) + 2 * gap_{s-1} + \left(A_{ufast,s-1} - D_{uslow,s}\right)\right) :\ t \in T_1, u \in T_2$
mainLB	$= \max_s(mainLB_s)$
$stationLB_s$	Lower bound on *Interval* obtained by assigning train types to platforms in station s (s = 1 to S)
stationLB	$= \max_s$ (s=1 to S) $\{stationLB_s\}$
maxLB	Maximum lower bound on optimal value of *Interval* (= max{*mainLB*, *stationLB*}) (real, > 0)
$lowIntMain_{tus}$	Lowest interval of train type t that must be compared to the "interval 0 of train type u" to ensure headway constrains are enforced on the main line between station s and s+1 (integer) (defined $\forall s$ from 0 to S, $\forall (t,u)$: $t < u$ and both train t and u use the portion of the main line between station s and s+1)

Table 5.1 Continued

$highIntMain_{tus}$	[replace "Lowest" with "Highest" in the above definition]
$lowIntPlat_{tup}$	Lowest interval of train type t that must be compared to the "interval 0 of train type u" to ensure headway constrains are enforced on platform p (integer) ($\forall(t,u)$: train types t and u are eligible to visit platform p defined and $t < u$)
$highIntPlat_{tup}$	[replace "Lowest" with "Highest" in the above definition]
Decision Variables:	
Interval	Interval duration (minutes) (real, > 0)
A_{ts}	Arrival time of original train of type t at station s (real, > 0)
D_{ts}	Departure time of original train of type t from station s (real, ≥0)
$Z_{itsu} = \begin{cases} 1 \\ 0 \end{cases}$	If "interval i train of type t" appears on the portion of the main line between station s and s+1 **before** original (i.e. interval 0) train of type u appears there Otherwise (binary) (define $\forall s, t, u$ for which $lowIntMain_{tus}$ is defined, $\forall i$: $lowIntMain_{tus} \le i \le highIntMain_{tus}$)
$Z'_{itsu} = \begin{cases} 1 \\ 0 \end{cases}$	If "interval i train of type t" uses the portion of the main line between station s and s+1 **before** original (i.e. interval 0) train of type u uses it Otherwise (binary) ($\forall s$: $0 \le s \le S$, $\forall (t,u$: $t < u$, $d_t \ne d_u)$ $oStation_t \le s < dStation_t$ and $dStation_u \le s \le oStation_u$, $\forall i$: $lowIntMain_{tus} \le i \le highIntMain_{tus}$)
$Q_{ts} = \begin{cases} 1 \\ 0 \end{cases}$	If train type t stops in station s Otherwise (binary) (same index ranges as for $dMin_{ts}$)
$X_{tp} = \begin{cases} 1 \\ 0 \end{cases}$	If train type t is assigned to platform p Otherwise (binary) (t = 1 to T, p = 1 to P)
$Y_{itpu} = \begin{cases} 1 \\ 0 \end{cases}$	If "interval i train of type t" uses platform p **before** original (i.e. interval 0) train of type u uses it Otherwise (binary) (defined $\forall p, t, u$ for which $lowIntPlat_{tup}$ is defined, $\forall i$: $lowIntPlat_{tup} \le i \le highIntPlat_{tup}$)
V_{itpu}=	[replace "before" with "after" in the above definition]

Table 5.2 Equivalent mathematical expressions for index ranges mentioned in Table 5.1

Index ranges:	
English description	Equivalent mathematical expression
Train type t uses the portion of the main line between stations s and s+1	($\forall t, \forall s$: $1 \le t \le T_1$ and $oStation_t < s < dStation_t$ OR $T_1+1 \le t \le T$ and $dStation_t < s < oStation_t$)
Both train t and u use the portion of the main line between station s and s+1	$d_t = d_u$ and $d_t = 1$and $oStation_t \le s < dStation_t$ and $oStation_u \le s < dStation_u$) OR $d_t = 2$ and $dStation_t \le s < oStation_t$ and $dStation_u \le s < oStation_u$)
Both train t and u are eligible to visit platform p	$d_t = d_u = 1$and $oStation_t < s_p < dStation_t$ and $oStation_u < s_p < dStation_u$) OR $d_t = d_u = 2$ and $dStation_t < s_p < oStation_t$ and $dStation_u < s_p < oStation_u$) OR $d_t = 1$ and $d_u = 2$ and $oStation_t < s_p < dStation_t$ and $dStation_u < s_p < oStation_u$)
Train type t is eligible to visit platform p	$\forall p$, t : $forbid_{t(s\_pi)} = 0$
Train type t arrives at station s	($\forall t, \forall s$: $1 \le t \le T_1$ and $oStation_t < s \le dStation_t$ OR $T_1+1 \le t \le T$ and $dStation_t \le s < oStation_t$)
Train tope t departs from station s	($\forall t, \forall s$: $1 \le t \le T_1$ and $oStation_t \le s < dStation_t$ OR $T_1+1 \le t \le T$ and $dStation_t < s \le oStation_t$)

Table 5.3 Parallel machine scheduling problem for computing *stationLB$_s$*

Indices	
t	train type ($1 \le t \le T$)
p	platform ($1 \le p \le P$)
Parameters	
s	Station under consideration.
T	Number of train types stopping in station s
P	Number of platforms in station s
$dMin_t$	Minimum required dwell time for train type t in the station (= $dMin_{ts}$ in Table 5.1, real, > 0)
$hPlatform_p$	Lower bound on inimum required headway between trains stopping on platform p = min{$hPlatforSame_p$, $hPlatformDiff_p$} (real, ≥ 0)
Decision variables	
X_{tp}	= 1 if train type t is assigned to platform p (binary).
$stationLB_s$	Minimum makespan for the machine scheduling problem (real, > 0).
Math program	

Objective:

minimize $stationLB_s$

Subject to:

$$\sum_{p=1}^{P} X_{tp} = 1 \quad \forall t$$

$$\sum_{t=1}^{T} (X_{tp})(dMin_t + hPlatform_p) \le stationLB_s \quad \forall p$$

5.3.1. Mathematical Model

There are nine decision variables in the model: the first three are real variables and other six are yes-no decision variables forming our mixed integer linear program. As mentioned in Section 1, *Interval* is the length in minutes of the dispatching cycle. It is the quantity we seek to minimize in this study. A_{ts} and D_{ts} are the arrival and departure times of the original train of type t at station s. The binary variables Q_{ts} determine whether or not train type t stops at a station. The binary variables X_{tp} indicate which train types are assigned to which platforms. The binary variables Z_{itsu} indicate the sequence of train types on the main line that move in the same direction; while, Z'_{itsu} indicate the sequence of train types on the main line that move in opposite directions. The binary variables Y_{itpu} and V_{itpu} indicate the sequence of train types visiting the platforms.

Our MILP formulation of this problem is as follows:

Minimize $$a_1 \times Interval + a_2 \times \sum_{t \in T} \left(A_{t,dStation_t} - D_{t,oStation_t} \right) \qquad (5\text{-}1)$$

Subject to

$$0 \le D_{t,oStation_t} \le Interval \qquad \forall t \in T \qquad (5\text{-}2)$$

$$D_{ts} + travMin_{ts} \le A_{t,s+1} \qquad \forall t : d_t = 1, \forall s : oStation_t \le s < dStation_t \qquad (5\text{-}3)$$

$$D_{ts} + travMin_{t,s-1} \le A_{t,s-1} \qquad \forall t : d_t = 2, \forall s : dStation_t < s \le oStation_t \qquad (5\text{-}4)$$

$$\left(D_{ts} - A_{ts} \right) \ge dMin_{ts} \qquad \forall (t,s) : dMin_{ts} \text{ is defined} \qquad (5\text{-}5)$$

$$\left(D_{ts} - A_{ts} \right) = 0 \qquad \forall (t,s) : forbid_{ts} \text{ is defined and } forbid_{ts} = 1 \qquad (5\text{-}6)$$

$$\left(A_{t,dStation_t} - D_{t,oStation_t}\right) \le jMax_t \qquad \forall\, t \in T \qquad (5\text{-}7)$$

Constraints (5-8) – (5-11) are created $\forall i,t,s,u$ for which Z_{itsu} is defined and $d_t = d_u = 1$

$$\left(i \times Interval + D_{ts}\right) - D_{us} - \left(1 - Z_{itsu}\right)M \le -hTrack_s \qquad (5\text{-}8)$$

$$\left(i \times Interval + D_{ts}\right) - D_{us} + Z_{itsu}M \ge hTrack_s \qquad (5\text{-}9)$$

$$\left(i \times Interval + A_{t,s+1}\right) - A_{u,s+1} - \left(1 - Z_{itsu}\right)M \le -hTrack_s \qquad (5\text{-}10)$$

$$\left(i \times Interval + A_{t,s+1}\right) - A_{u,s+1} + MZ_{itsu} \ge hTrack_s \qquad (5\text{-}11)$$

Constraint (5-12) needs slightly different range than constrains (5-8) – (5-11).

$$Z_{itsu} \le Z_{i-1,t,s,u} \qquad (5\text{-}12)$$

Constraints (5-13) – (5-16) are created $\forall i,t,s,u$ for which Z_{itsu} is defined and $d_t = d_u = 2$

$$\left(i \times Interval + D_{t,s+1}\right) - D_{u,s+1} - \left(1 - Z_{itsu}\right)M \le -hTrack_s \qquad (5\text{-}13)$$

$$\left(i \times Interval + D_{t,s+1}\right) - D_{u,s+1} + MZ_{itsu} \ge hTrack_s \qquad (5\text{-}14)$$

$$\left(i \times Interval + A_{ts}\right) - A_{us} - \left(1 - Z_{itsu}\right)M \le -hTrack_s \qquad (5\text{-}15)$$

$$\left(i \times Interval + A_{ts}\right) - A_{us} + MZ_{itsu} \ge hTrack_s \qquad (5\text{-}16)$$

Constraint (5-17) needs slightly different range than constrains (5-3) – (5-16).

$$Z_{itsu} \le Z_{i-1,t,s,u} \qquad (5\text{-}17)$$

Constraints (5-18) and (5-19) are created $\forall i,t,s,u$ for which Z'_{itsu} is defined and $d_t = 1$ and $d_u = 2$

$$\left(i \times Interval + A_{t,s+1}\right) - D_{u,s+1} - \left(1 - Z'_{itsu}\right)M \leq -gap_s \tag{5-18}$$

$$\left(i \times Interval + D_{ts}\right) - A_{us} + MZ'_{itsu} \geq gap_s \tag{5-19}$$

Constraint (5-20) needs slightly different range than constrains (5-18) and (5-19).

$$Z'_{itsu} \leq Z'_{i-1,t,s,u} \tag{5-20}$$

$$Q_{ts} \leq M \times \left(D_{ts} - A_{ts}\right) \qquad \forall t, \forall s : Q_{ts} \text{ is defined} \tag{5-21}$$

$$D_{ts} - A_{ts} \leq M \times Q_{ts} \qquad \forall t, \forall s : Q_{ts} \text{ is defined} \tag{5-22}$$

$$\sum_{p \in P_s} X_{tp} = Q_{ts} \qquad \forall t, \forall s : Q_{ts} \text{ is defined} \tag{5-23}$$

$$X_{tp} + X_{up} - 1 \leq Y_{itpu} + V_{itpu} \tag{5-24}$$

$$X_{tp} \geq Y_{itpu} + V_{itpu} \tag{5-25}$$

$$X_{up} \geq Y_{itpu} + V_{itpu} \tag{5-26}$$

$$A_{u,s_p} - \left(i \times Interval + D_{t,s_p}\right) + \left(1 - Y_{itpu}\right)M \geq hPlatformSame_p \tag{5-27}$$

$$\left(i \times Interval + A_{t,s_p}\right) - D_{u,s_p} + \left(1 - V_{itpu}\right)M \geq hPlatformSame_p \tag{5-28}$$

$\forall p,t,u,i$ for which Y_{itpu} is defined and $d_t = d_u$

$$A_{u,s_p} - \left(i \times Interval + D_{t,s_p}\right) + \left(1 - Y_{itpu}\right)M \geq hPlatformDiff_p \tag{5-29}$$

$$\left(i \times Interval + A_{t,s_p}\right) - D_{u,s_p} + \left(1 - V_{itpu}\right)M \geq hPlatformDiff_p \tag{5-30}$$

$$\forall p,t,u,i \text{ for which } V_{itpu} \text{ is defined and } d_t = 1 \text{ and } d_u = 2$$

$$\begin{cases} Y_{itpu} \leq Y_{i-1,t,p,u} \\ V_{itpu} \leq V_{i+1,t,p,u} \\ Y_{itpu} \leq Z_{i,t,s_p,u} \\ V_{itpu} \leq 1 - Z_{i,t,s_p,u} \end{cases} \tag{5-31}$$

$$Interval \geq \left(D_{t,s_p} - A_{t,s_p}\right) + X_{tp} \times hPlatformSame_p \tag{5-32}$$

$$\forall p, \left(\forall t : \left(\left(oStation_t < s_p < dStation_p, d_t = 1\right)\right.\right.$$

$$\left.\left.\text{or } \left(dStation_t < s_p < oStation_t, d_t = 2\right), forbid_{t,s_p} = 0\right)\right)$$

$$maxLB \leq Interval \leq minUB \tag{5-33}$$

The objective function (5-1) is a weighted sum of the primary objective, *Interval*, and the secondary objective, total journey time of all train types combined, that we want to minimize. The first objective is the main focus of this study, so $a_1 >> a_2$ in most experiments. Constraint (5-2) ensures that the original train of each type departs its starting point during the first interval (i.e. “interval 0”). Constraint (5-3) guarantees the sequencing of stations that each eastbound train type should visit and also requires that at least $travMin_{ts}$ be the traveling time for train type t along the main line between station s and s+1. Constraint (5-4) is the same as constraint (5-3) but for westbound trains. Constraints (5-5) and (5-6) ensure that (i) each train type can stop for at least the required minimum amount of time in each station it visits if $forbid_{ts} = 0$ and (ii) train type t does not spend any time in station s if $forbid_{ts} = 1$. Constraint (5-7) ensures that the total journey time of train of type t does not exceed the maximum allowed value, $jMax_t$.

Constraints (5-8) and (5-9) are disjunctive constraints that enforce the headway restriction on two eastbound trains of type *t* and of type *u* departing station *s* towards station *s*+1 for all *s* from 0 to *S*. In particular, these constraints guarantee that the interval *i* train of type *t* appears at the merge point just after station *s* either at least $hTrack_s$ minutes before (5-8) or after (5-9) the original train of type *u* appears there for all *s* from 0 to *S*, for all pairs of train types that travel along the portion of the main line between station *s* and *s*+1, and for all intervals *i* of train type *t* that could possibly interfere with the original train of type *u*. Note that, although constraints (5-8) and (5-9) consider a finite number of trains of type *t* and only one train of type *u*, they enforce headway constraints on the main track for *all* trains of type *t* versus *all* trains of type *u* owing to the repetitive, cyclic nature of the timetable. The idea is the same as the second problem considered in Chapter 4 which is due to repetitive cyclic fashion of problem, every pair-wise comparison can move over planning horizon. Constrains (5-10) and (5-11) are ordering constraints that work together with constraints (5-8) and (5-9) to separate eastbound train type *t* and *u* at the arriving point of station *s*+1 coming from station *s*. Based on these constraints, if train type *t* leaves station *s* at least $hTrack_s$ minutes before (5-8) the original train of type *u*, it should arrive at station *s*+1 at least $hTrack_s$ minutes before (5-10) the original train of type *u*. In a similar way, if train type *t* leaves station *s* at least $hTrack_s$ minutes after (5-9) the original train of type *u*, it should arrive at station *s*+1 at least $hTrack_s$ minutes after (5-11) the original train of type *u*. In other words, constraints (5-10) and (5-11) guarantee the headway on the main line and ensure that trains do not overtake due to different speeds while moving on the main line between station *s* and *s*+1. Regarding the ordering of trains on the main line, constraint (5-12) states that if the interval i train of type t is before the original train of type u, then the interval i-1 train of type t must also be before the original train of type u. Constraints (5-13) – (5-17) are the counterparts of the constraints (5-8) – (5-12) that are defined for westbound train types. Constraints (5-18) and (5-19) are disjunctive constraints that enforce the time separation between two train types moving in opposite directions on the portion of the main line between station s and s+1 for all s from 0 to S. In particular, these constraints guarantee that the interval i train of type t uses the portion of the main line between station s and s+1 either at least gap_s minutes before (5-18) or after (5-19) the original train of type u uses that portion for all s from 0 to S, for all pair of train types that travel along that portion of the main line such that trains of type t that are eastbound and trains of type u that are westbound. Constraint (5-20) states that if the interval i train of type t is before the original train of type u, then the interval i-1 train of type t must also be before the original train of type u. Constraint (5-21) states that if there is no difference between arrival time and departure time of train type t in station s, then train type t must not stop in station s. On the other hand, Constraint (5-22) states that if there is a difference between arrival time and departure time of train type t in station s, then train type t must stop in station s. Constraint (5-23) ensures that each train type stopping (not stopping) in a station visits 1 (0) a platform in the station. Constraints (5-24) – (5-26) ensure that, if two train types, regardless of their directions, utilize the same platform, then the first train type must either use the platform before or after the second train type; the two train types cannot utilize the platform simultaneously. Constraints (5-27) – (5-30) enforce the headway restriction on all stations platforms. In particular, constraints (5-27) and (5-28) are disjunctive constraints and guarantee the headway separation between two train types traveling in the same direction. In other words, if both train types *t* and *u* can stop at station *s* where platform *p* resides and use platform *p*, these constraints guarantee that the interval *i* train type *t* uses platform *p* either at least $hPlatformSame_p$ minutes before (5-27) or after (5-28) the original train of type *u* uses it for all *p* and for all intervals *i* of train type *t* that could possibly interfere with the original train type *u* at that platform. Again, due to repetitive, cyclic nature of the model, constraints (5-27) and (5-28) enforce headway constraints on the station platform for *all* trains of type *t* versus *all* trains of type *u* owing to the repetitive, cyclic nature of the timetable. Constraints (5-29) and (5-30) are disjunctive constraints that guarantee the platform headway separation between an eastbound train type *t* and a westbound train type *u*, if both stop and reside on platform *p* in station *s*. According to these constraints, if both trains stop on platform *p*, then the interval *i* train of type *t* uses platform *p* either at least $hPlatformDiff_p$ minutes before (5-29) or after (5-30) the original train of type *u* uses it for all *p* and for intervals *i* of train type *t* that could possibly interfere with the original train of type *u* at that platform. The first two constrains in (5-31) state that if the interval *i* train of type *t* is before (after) the original train of type *u*, then the interval *i*-1 (*i*+1) train of type *t* must also use platform *p* before (after) the original train of type *u*. The second two constraints in (5-31) ensure that the ordering of two trains on the main line between station *s* and *s*+1 agrees with the ordering in which these trains visit the same platform in station *s*. Constraint (5-32) guarantees that the cycle length *Interval* is large enough so that each stop by a train type at a platform can be made without the train overlapping with its sister train from the next interval. Constraint (5-33) forces *Interval* to take a value that is no lower than its lower bound and no higher than its upper bound; these bounds are defined and discussed in Sections 5.3.2 and 5.3.3.

5.3.2. Upper bound on optimal value

We now describe the calculation of minimum upper bound used in the mathematical model. The parameter *minUB* provides an upper bound on the optimal value of *Interval*. It equals the lowest value of *Interval* among the T_1! feasible orderings of eastbound trains and T_2! feasible orderings of westbound trains. This is obtained by considering all possible T_1! cyclic orderings of eastbound train types and then scheduling them in order one-at-a-time such that (a) all train types achieve their minimum station dwell and traveling time; (b) there is no passing; and (c) no two train types may be in the same station at the same time. During the scheduling process, the first eastbound train type is scheduled so it departs its origin at time 0. Then, each subsequent eastbound train type is scheduled one-at-a-time as early as possible considering the above assumptions and the fact that the subsequent train must be separated by $hPlatformSame_p$ within stations and $hTrack_s$ on the main line from its predecessor. Then we update $minUB = Interval + gap_s$, where *Interval* is the difference between arriving the last eastbound train at its destination and departing the first train from its origin. At the next step we do the same procedure for the T_2! of westbound train types.

5.3.3. *Lower bound on optimal value*

Parameter *maxLB* provides a lower bound on the optimal value of *Interval*. It equals the higher of two different lower bounds: *stationLB* and *mainLB*. Parameter *stationLB* is the minimum value of *Interval* that could possibly be achieved by considering train dwell time, headway in stations and the number of tracks in each station or station capacity. It is obtained by solving a parallel machine scheduling problem for each station *s*, where platforms are considered as machines and train types that stop in station *s* as jobs with processing time $dMin_{ts}$ with $hPlatformSame_p$ as setup time (Table 5.3). The objective of this machine scheduling problem is to minimize makespan. Let $stationLB_s$ be the optimal makespan of the machine scheduling problem related to station *s*. Then $stationLB = \max_s\{stationLB_s\}$.

Parameter *mainLB* is the minimum value of *Interval* that could possibly be achieved due to headway constraints on the main line only. Let $mainLB_s$ be the lowest possible value of *Interval* due to headway restrictions on the portion on the portion of the main track between station *s* and *s*+1 for all *s* from 0 to *S*+1. Then $mainLB = \max_s\{mainLB_s\}$. In order to obtain $mainLB_s$, first we identify the set of eastbound trains that use the portion of the main line between station *s* and *s*+1and we sort these train types in increasing order of travel time on the main line between station *s* and *s*+1. Then we find the difference between the arrival of the last (slowest) train at s+1 and the departure of the first (fastest) train from *s*.

This partial lower bound should be then increased by the gap_s. Up to this point we have only considered eastbound train types. In the similar way, we identify the set of westbound trains that use the portion of the main line between stations *s*+1 to *s* and sort them in increasing order of travel time along this portion of the main line; and calculate the difference between arrival of the slowest train to station *s* and departure of fastest train from station *s*+1, then update $mainLB_s$ by adding this value. Finally increasing $mainLB_s$ by gap_s will provide us with the final value of $mainLB_s$.

5.3.4. *Derivation of secondary parameters*

We now describe the derivation of the secondary parameters. The secondary parameters $lowIntMain_{tus}$, $highIntMain_{tus}$, $lowIntPlat_{tup}$, and $highIntPlat_{tup}$ allow us to construct required number of constraints enforcing the headway restriction on each portion of the main line and on each platform depending upon whether or not a pair of train types uses that portion of main line or platform. This is also crucial because of the cyclic nature of the timetable. Here we define the "interval *i* train of type *t*" to be the train of type *t* that departs its starting point during interval *i* – i.e. sometime between time (i) × *Interval* and time (i+1) × *Interval* – where *i* is any negative or non-negative integer. In general, the above parameters indicate how many "intervals worth" of trains of a given type could possibly have a headway conflict with the original train – i.e. the train dispatched during interval 0 – of another type on a certain station platform or along a certain portion of the main line where a headway constraint is required. The computation of these four parameters is based on and begins with a calculation of the earliest and latest possible arrival/departure time of the original train of each type at/from each station (including departing from origin and arriving at destination). The earliest possible arrival and departure times assume each train starts from origin at time 0 and spends minimum dwell time ($dMin_{ts}$) at each station.

The latest possible arrival and departure times are calculated by first assuming that each train type *t* (regardless of its direction) departs at time *minUB* and makes a journey of duration $jMax_t$; backward recursion is used to make the arrival and departure times as high as possible still adhering to minimum main line traveling times and minimum required station dwell times.

5.3.5. Determining the smallest possible value of Big M

In this problem we have a very large number that is used for ordering (either-or) constraint. Usually an arbitrary large number is picked up for the binary coefficients in either-or constraints. Because this value is very problem dependent, the question is how large this coefficient should be for each specific problem. For our general train timetabling and platforming problem, the same method as discussed in Chapter 4 is applied. In other words, one of the ordering constraints is selected and based on other parameters, a value is determined so as to keep the problem feasible for all possible combinations of trains at all stations and platform (without loss of generality, it is possible to consider all ordering constraints and choose the maximum value of M as the coefficient. This, of course, requires more pre-processing and may increase the computational time).

5.3.6. A note on problem complexity

We now prove a theorem regarding the problem complexity.

Theorem 1. The train timetabling and routing optimization problem defined by the MILP formulation in Section 5.3.1 is NP-hard.

Proof. The decision problem $P||C_{max}$, the parallel machine scheduling problem with makespan minimization, is known to be NP-complete. This problem is polynomially reducible to the train timetabling and routing problem. Consider a train problem instance where $S = 1$; therefore $oStation_t = 0$ for all t; $dStation_t = 2$ for all t; $P = P$ (the number of platforms in the train problem equals the number of machines in the machine scheduling problem); T equals the number of jobs in $P||C_{max}$; $travMin_{ts} = 0$ for all t, and for $s = 0$ and 1; $forbid_{t1} = 0$ for all t; $dMin_{t1}$ equals the processing time of job t in $P||C_{max}$ for all t; $jMax_t$

= $dMin_{t1}$ for all t; $hTrack_s$ = gap_s = 0 for s = 0 and 1; $hPlatformSame_p$ = $hPlatformDiff_p$ = 0 for all p; a_1 = 1; and a_2 = 0. In other words, the problem is to assign T train types to P platforms in a single station disregards all other problem aspects. In this problem, *Interval* minimization is equivalent to makespan minimization with P parallel machine and T jobs, shown symbolically as $P||C_{max}$. Since $P||C_{max}$ is polynomially reducible to the problem at hand and $P||C_{max}$ is NP -complete, it follows that that the problem is NP -complete. And this completes the NP-hardness proof.■

5.4. Two illustrative examples

The mixed integer program was coded into Microsoft Visual C++ 2008. IBM ILOG Concert Technology was used to define the model within C++ and call the MILP solver IBM ILOG CPLEX 12.4 to solve instances defined in text files. The code includes procedures for automatically computing *minUB*, *mainLB*, $stationLB_s$, *stationLB*, *maxLB*, $lowIntMain_{tus}$, $highIntMain_{tus}$, $lowIntPlat_{tup}$, and $highIntPlat_{tup}$ before the constraints are constructed. The computation of $stationLB_s$ for all s involves calling CPLEX to solve the parallel machine scheduling problem, as discussed before, for each station.

Before presenting illustrative examples, let us discuss the default setup used for all problem instances in this chapter unless otherwise specified. This setup reflects the fact that the primary objective in the mathematical model is to minimize the cycle length and secondary, subordinate objective is to minimize the total train journey time. To accomplish this, all primary parameters besides a_1 and a_2 are multiples of 0.5. The objective function weights are determined in such a way that the primary goal of minimizing the cycle length does not interfere with the secondary one, but is large enough to be able to identify, among all solutions tied for having minimum *Interval*, a solution that ties for having the smallest total journey time for all train types combined. As shown in Chapter 3 this way we can select an optimal solution from a set of alternative optimal solutions. Therefore, in majority of experiments it is assumed that $a_1 = 1$ and $a_2 = 0.0001$. In this case, if $\sum_{t=1}^{T} jMax_t \leq 4999.5$, then the maximum possible value of the second term of the objective function is 0.49995 and 0.5 is the minimum change in the value of the first term of the objective function. Therefore, these two values never interfere with each other.

We now present two illustrative problem instances using above setup and discuss their optimal solutions. Text files defining all problem instances described here are available from the authors upon request.

The input data for the first problem instance is given in Table 5.4. This problem instance considers four train types with different stopping frequencies. Train types 1 and 2 eastbound trains, and train types 3 and 4 are westbound trains. The primary parameters, except s_p and p_s which can be inferred from the data, are shown in Table 5.4. Table 5.5 shows an optimal solution for this instance in the form of a cyclic timetable. This solution was obtained in 131 seconds. In this table, the platform assignments are shown at the top; actual dwell time in each station for each train type is shown in the next, the actual traveling times on each portion of the main line follow; and detailed schedule for the first three cycles is displayed. The schedule of each train can be derived from this table using arrival and departure times of the train at each station. Figure 5.3 shows the same optimal schedule in the form of time-space diagram. The platform assignments in stations 3 and 5 are indicated by slight differences in the vertical placement of the horizontal line. As can be seen from Table 5.5 and Figure 5.3, the optimal value for this instance, i.e. the minimum value of *Interval*, is 36 minutes. This value is strictly greater than the lower bound *maxLB* (= 12) and strictly less than the upper bound *minUB* (= 83). In this particular example actual total station dwell times are {12, 33.5, 8, 19.5} for train types 1, 2, 3 and 4 respectively. The total journey times are {34, 55.5, 32, 43.5} for train types 1, 2, 3 and 4 respectively which are strictly less than the maximum allowed values of 100, 110, 110 and 100, respectively.

Figure 5.4 shows an optimal time-space diagram for a second problem instance with T = 6 train types, where $T_1 = \{1, 2, 3\}$ and $T_2 = \{4, 5, 6\}$, and S = 20 stations. In this example we assume 3 train types are eastbound trains and the other 3 train types are westbound trains. We further assume that all train types are long distance trains, mathematically $oStation_t = 0$ and $dStation_t = 21$, $\forall\ t \in T_1 \wedge d_t = 1$, and $oStation_t = 21$ and $dStation_t = 0$, $\forall\ t \in T_2 \wedge d_t = 2$. The optimal value of *Interval* is 75 minutes. In Figure 5.4 two complete cycles or intervals are depicted. As can be seen from this figure, some trains slow down at some links between stations, or wait longer time at some stations in order to create room for passing possibility. These passing possibilities will increase problem flexibility which will be resulted in an optimal value for *Interval*.

Table 5.4 Input data for illustrative example #1

T_1	T_2	S	P	$hTrack_s$	gap_s	$hPlatformSame_p$	$hPlatformDiff_p$	a_1	a_2	$forbid_{ts}$
2	2	6	8	1 for all s	1 for all s	0.5 for all p	1 for all p	1	0.0001	0 for all t and s

	West Terminal	Stat. 1	Stat. 2	Stat. 3	Stat. 4	Stat. 5	Stat. 6	East Terminal	$jMax_t$
#Platforms	-	1	1	2	1	2	1	-	
$dMin_{1s}$	Origin	1	3	3	0	2	3	Destination	100
$dMin_{2s}$	Origin	1	2	4	1	5	4	Destination	110
$dMin_{3s}$	Destination	1	1	1	1	1	2	Origin	110
$dMin_{4s}$	Destination	5	1	3	2	4	0	Origin	100
$travMin_{1,s-1}$		3	3	3	3	3	3	4	
$travMin_{2,s-1}$		3	3	3	3	3	3	4	
$travMin_{3,s-1}$		3	3	3	3	3	3	4	
$travMin_{4,s-1}$		3	3	3	3	3	3	4	

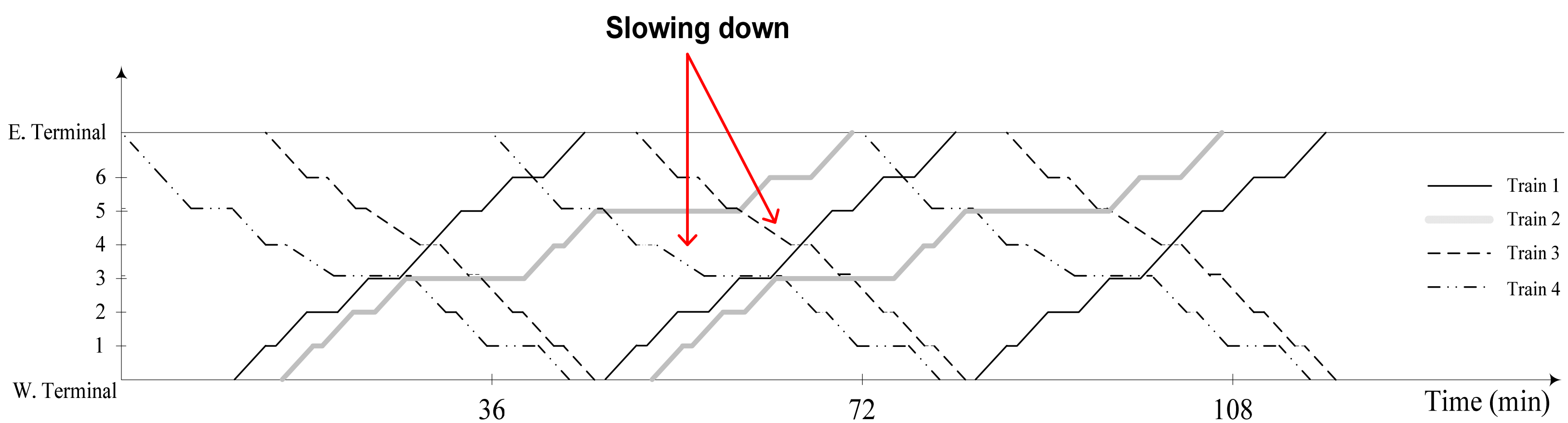


Figure 5.3 Optimal time-space diagram for illustrative example # 1.

Table 5.5 Optimal platform assignment and timetable for illustrative example #1

Minimum *Interval* = 36		Stat. 1	Stat. 2	Stat. 3		Stat. 4	Stat. 5		Stat. 6
		Plat. 1	Plat. 2	Plat. 3	Plat. 4	Plat. 5	Plat. 6	Plat. 7	Plat. 8
Platform Assignment	Train 1	1	1	1	0	0	0	1	1
	Train 2	1	1	1	0	1	0	1	1
	Train 3	1	1	0	1	1	1	0	1
	Train 4	1	1	0	1	1	1	0	0

Actual station dwell time

	Stat. 1	Stat. 2	Stat. 3	Stat. 4	Stat. 5	Stat. 6	Total
Train 1	1	3	3	0	2	3	12
Train 2	1	2	11.5	1	14	4	33.5
Train 3	1	1	1	2	1	2	8
Train 4	5	1	7.5	2	4	0	19.5

Actual traveling time from s to s+1 for eastbound trains and from s+1 to s for westbound trains

	E. Terminal	Stat. 1	Stat. 2	Stat. 3	Stat. 4	Stat. 5	Stat. 6
Train 1	3	3	3	3	3	3	4
Train 2	3	3	3	3	3	3	4
Train 3	3	3	3	3	5	3	4
Train 4	3	3	3	5	3	3	4

Timetable

	E. Terminal	Stat. 1	Stat. 2	Stat. 3	Stat. 4	Stat. 5	Stat. 6	W. Terminal
Train 1	11	14 – 15	18 – 21	24 – 27	30	33 – 35	38 – 41	45
Train 2	15.5	18.5 – 19.5	22.5 – 24.5	27.5 – 39	42 – 43	46 – 60	63 – 67	71
Train 3	46	42 – 43	38 – 39	34 – 35	29 – 31	23 – 24	18 – 20	14
Train 4	43.5	35.5 – 40.5	31.5 – 32.5	21 – 28.5	14 – 16	7 – 11	4	0
Train 1	47	50 – 51	54 – 57	60 – 63	66	69 – 71	74 – 77	81
Train 2	51.5	54.5 – 55.5	58.5 – 60.5	63.5 – 75	78 – 79	82 – 96	99 – 103	107
Train 3	82	78 – 79	74 – 75	70 – 71	65 – 67	59– 60	54 – 56	50
Train 4	79.5	71.5 – 76.5	67.5 – 68.5	57 – 64.5	50 – 52	43 – 47	40	36
Train 1	83	86 – 87	90 – 93	96 – 99	102	105 – 107	110 – 113	117
Train 2	87.5	90.5 – 91.5	94.5 – 96.5	99.5 – 111	114 – 115	118 – 132	135 – 139	143
Train 3	118	114 – 115	110 – 111	106 – 107	101 – 103	95 – 96	90 – 92	86
Train 4	115.5	107.5 – 112.5	103.5 – 104.5	93 – 100.5	86 – 88	79 – 83	76	72

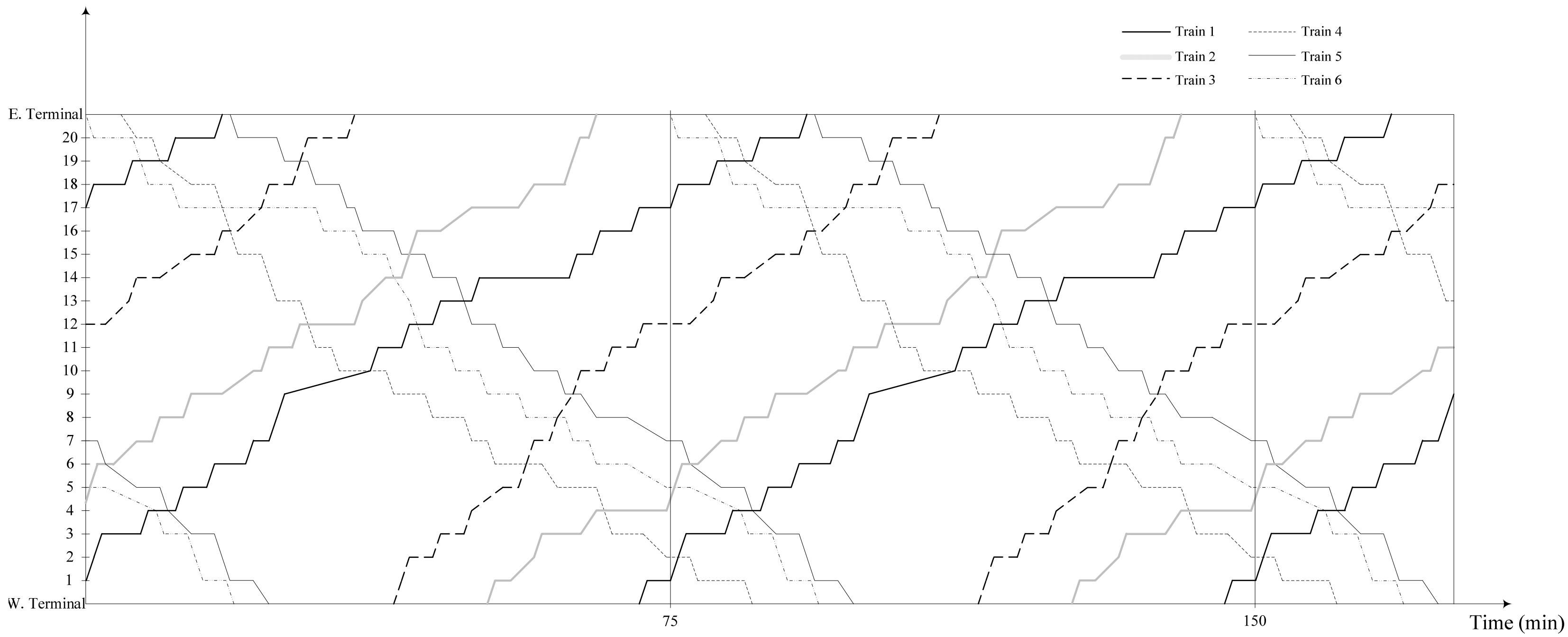


Figure 5.4 Optimal time-space diagram for illustrative example # 2.

5.5. Additional experiments: setup, results, and discussion

We now perform several additional experiments to confirm the effectiveness of the model presented in Sections 5.3.1. These experiments consider hundreds of problem instances, most solved to optimality. The results in this section are obtained by CPLEX 12.4 running on an IBM-compatible desktop computer using an Intel Core 2 Duo processor with two 2.83 GHz cores, and 2 GB of RAM. In all experiments in this section, CPLEX 12.4 is given 10000 minutes time limit for solving the math program. This time is in addition to the time used by CPLEX to solve preliminary machine scheduling problems in each station and any other time needed to set up the problem. The problem instances are defined by the values of the primary input parameters given in Table 5.6.

Now we will discuss eight sets of experiments each showing one aspect of the train timetabling problem presented in this paper by the mathematical program.

Table 5.7 shows the results from the first set of experiments in which 28 problem instances are considered. The main parameters defining each instance are on the left and the main aspects of the solution are given on the right. Also $hTrack_s$, gap_s, $hPlatformSame_p$ and $hPlatformDiff_p$ are fixed at 1, 1, 0.5 and 1, unless otherwise specified. The results show that 27 (24) out of 28 problems are solved to optimality within time limit when both objectives are (only the first objective is) considered. The second last column in Table 5-7 shows the runtime required to find the optimal *Interval* for each problem instance when $a_1 = 1$ and $a_2 = 0$. The result in this column show there is an increase in runtime for 21 problem instances when the second objective is disregarded. The results indicate that the problem becomes more difficult to solve when the second objective is disregarded.

Table 5.6 Parameter values considered in Section 5.5

Parameter	Value
T	2,3,4,5,6
S	Integers ≥ 4 and ≤ 20
P	$S + X$ where $X \sim \mathbf{b}(S, 0.2)$ ($\mathbf{b}$ = binomial distribution)
$\lvert p_s \rvert$	1 or 2
$oStation_t$	0 or S+1
$dStation_t$	0 or S+1
$forbid_{ts}$	Usually = 1 if $\lvert p_s \rvert$; usually = $\mathbf{b}(0.5)$ if $\lvert p_s \rvert = 1$
$dMin_{ts}$	1, 2, 3, 4, 5, 6 (if $forbid_{ts} = 0$) (=0 if $forbid_{ts}$ =1)
$jMax_t$	$1.5 \times \sum dMin_{ts} + \sum travMin_{ts}$ + (random integer ≥ 40 and ≤ 60)
$hTrack_s$	1, 2, 3, 4 (held constant on entire main line in each instance)
gap_s	1, 2, 3, 4 (held constant on entire main line in each instance)
$hPlatformSame_p$	0.5
$hPlatformDiff_p$	1
(a_1, a_2)	(1, 0.0001)
$travMin_{ts}$	1, 2, 3, 4, 5, 6

Table 5.7 Results for 28 problem instances. All solutions are optimal unless indicated otherwise indicated.

Input parameters					Solution					
					Two objectives			First objective only		
Instance #	*S*	*P*	*T*	*maxLB*	*Interval*	Total jurney	Time (sec)	*Interval*	Time (sec)	change
1	5	5	3	12	67	102	7	Same	4	-3
2	6	8	4	16	45	175	22	Same	26	+4
3	9	10	3	13	31	203	10	Same	15	+5
4	9	11	4	13	37	270	290	Same	680	+390
5	10	10	4	13	42	251	26	Same	38	+12
6	10	11	3	12	45	195	22	Same	31	+9
7	10	11	6	30	64	335	2610	Same	1082	-1528
8	10	13	3	13	40	210	18	Same	15	-3
9	10	13	3	13	33	222	21	Same	16	-5
10	10	13	5	22	74	408	512	Same	744	232
11	12	13	3	13	36	231	33	Same	54	+21
12	12	13	4	13	29	193	1160	Same	29	4236
13	12	13	4	14	36	246	2771	Same	5790	+3019
14	12	13	6	26	79	413	10002*	56	10001*	-
15	12	14	4	13	23	224	4663	Same	6719	+2056
16	12	14	5	17	33	270	5258	Same	10001*	-
17	13	13	3	12	42	205	17	Same	24	+7
18	13	13	4	13	37	324	49	Same	58	+9
19	13	14	3	13	33	209	25	Same	26	+1
20	14	15	3	13	27.5	186	49	Same	52	+3
21	14	15	5	24	50	347	6056	Same	10002*	-
22	14	16	3	13	40	228	40	Same	38	-2
23	15	16	4	14	42	215	624	Same	17684	+17060
24	17	19	4	12	33	336	526	Same	1209	+683
25	17	19	5	17	46	398	767	Same	3546	+2779
26	18	19	4	12.5	40	261	335	Same	2004	+1669
27	20	20	6	26	75	510	8951	-	10000*	-
28	20	22	4	13	37	370	304	Same	623	+319

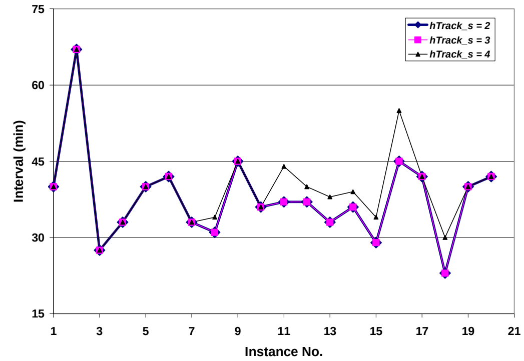


Figure 5.5 Effect of *hTrack$_s$* on *Interval* for the 20 problem instances

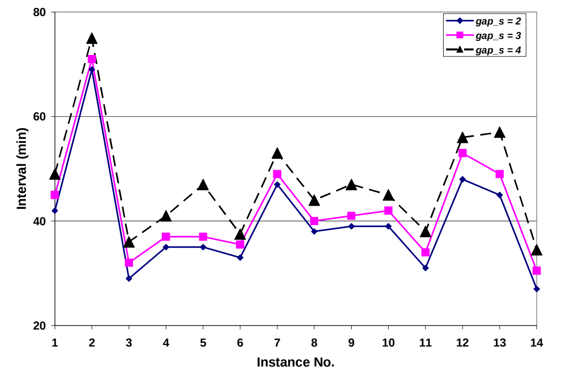


Figure 5.6 Effect of *gap$_s$* on *Interval* for the 14 problem instances

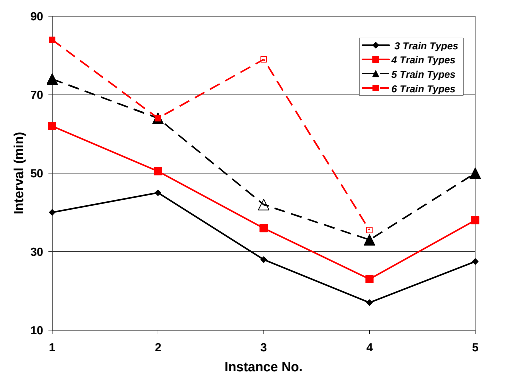


Figure 5.7 Impact of adding extra train types to each of 5 different problem instances.

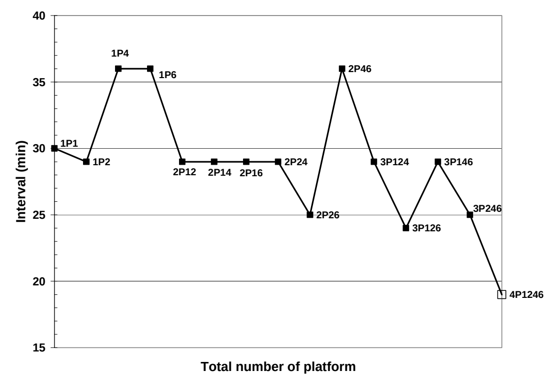


Figure 5.8 Impact on Interval of adding platforms to the stations of an existing line (labels indicated number of platform(s) added to station(s)).

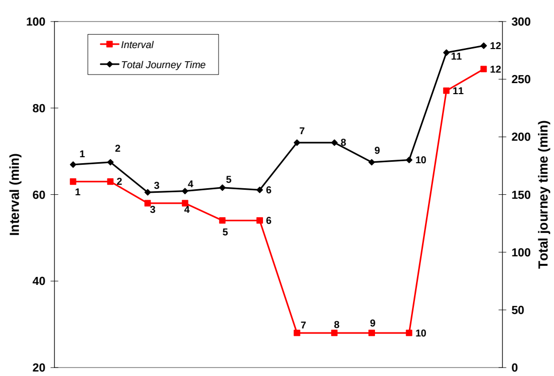


Figure 5.9 Effect of train heterogeneity on optimal *Interval*.

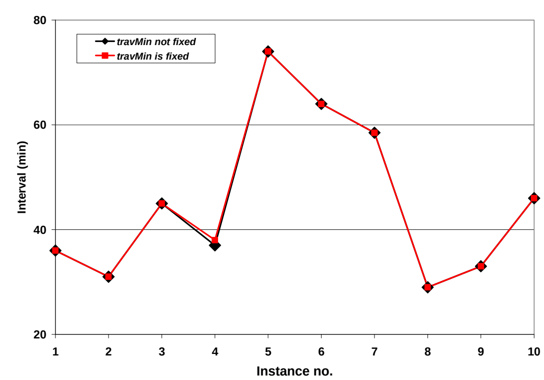


Figure 5.10 Effect of fixing *travMin* at its minimum value on *Interval*.

Figure 5.5 displays the results of a second set of experiments that show the impact of $hTrack_s$ on the optimal value of *Interval*. Here we consider three different values of $hTrack_s$ – 2, 3 and 4 – for 20 problem instances. Thus, we consider 60 problem instances total, and the same value of $hTrack_s$ is applied to the entire main line in each instance. The results show that there is no difference between $hTrack_s = 2$ and $hTrack_s = 3$, but optimal value of *Interval* may change (roughly 40 percent) when $hTrack_s$ is increased to 4. However, the amount of increase in *Interval* depends on the problem instance.

Figure 5.6 displays the results of a third set of experiments that show the impact of gap_s on the optimal value of *Interval*. For this experiment three different values of gap_s – 2, 3 and 4 – for 14 problem instances are considered. Therefore, 42 different instances are considered, and the same value of gap_s is applied to the entire main line in each instance. As can be seen from Figure 5.6, the optimal value of *Interval* increases as long as gap_s increases. This increase is mainly due to the fact that a train can use the track not only the track should be available, but also the next station should be available for either stop or passing.

Figure 5.7 displays the results of a fourth set of experiments that show how the optimal value of *Interval* changes when extra train types are added to a problem instance. For this experiment we considered 5 different instances with 3 train types each, and added on train type at a time up to 6 train type (T = 3, 4, 5 and 6). Thus, we consider 20 instances total ($10 \leq s \leq 14$ and $11 \leq p \leq 15$). Each base instance starts with three train types (T = 3) and extra train types are added to it one-by-one. That is, the input data for each instance with n train types (n = 4, 5, 6) is identical to the input data for the corresponding instance with n-1 train types except that one additional train types. The results show that the optimal value of *Interval* increases monotonically as new train types are added to an existing problem instance. However, the amount of increase in *Interval* depends on the base instance and on the exact specification of the added train type. In this experiment, in instance number 3 with $T = 5$ and 6, CPLEX could find a feasible solution within time limit. Also, CPLEX found a feasible solution for instance number 4 with $T = 6$. But, as can be seen from Figure 5.7, no feasible solution found for instance number 6 with $T = 6$ in the given time limit.

Figure 5.8 displays the results from a fifth set of experiments that show the impact on *Interval* of adding extra platform(s) to station(s) of an existing line. For this experiment, a base instance of 6 stations, 8 platforms and 4 trains (two eastbound train types and two westbound train types) in which stations 3 and 5 have two platforms and all other stations have one platform. We consider $\sum_{n=1}^{4} nC4 = 15$ possibilities of increasing the number of platforms from 8 to 16. Thus, we consider 15 total problem instances total. In this figure, the labels show the number of platforms and stations where these platforms are added. For example, 2P12 is an instance where 2 extra platforms are added to stations 1 and 2. The results show that capacity of the railway can be increased by adding more platforms to accommodate more train types, but it is important to consider which station can increase the capacity. This experiment also reveals how much the line capacity can be increased. In this particular case, the *Interval* can be decreased from 36 minutes to 19 minutes only if all stations have 2 platforms.

Figure 5.9 displays the results from a sixth set of experiments that show the effect of train heterogeneity on optimal *Interval* and total train journey time of an existing line. In this experiment, we consider 12 problem instances total each consists of 10 stations, 11 platform and 4 trains (two eastbound train types and two westbound train types) in 4 different categories. In Figure 5.9, the instances labeled by even numbers are the same as odd ones except their $minTrav_{ts}$ are swapped. In the first category (instances 1 and 2), one eastbound (westbound) train type is an express train that stops only at two stations and one eastbound (westbound) train type is a local train that stops at all stations. In the second category (i.e. instances 3-6), the first eastbound (westbound) train stops at the first half of the stations and the other eastbound (westbound) stops at the second half (instances 3 and 4). Then stopping pattern of these two eastbound (westbound) trains are swapped (instance 5 and 6). In the third category (i.e. instances 7-10), the first eastbound (westbound) train stops at odd stations, and the second eastbound (westbound) train stops at even stations (instance 7-8). Then in instances 9 and 10, the stopping patterns are swapped. In the last category (i.e. instances 11 and 12), all train types (either eastbound or westbound) are defined as local trains meaning that they stop at every station. Figure 5.9 shows that the *Interval* creates a concave function which has its minimum at category 3 where all trains stop at every other station. This means that in third category we create more passing options compared with other three categories. Hence, optimal value of *Interval* has been improved tremendously.

Figure 5.10 displays the results for the seventh set of experiments. In this experiment the effect of $travMin_{ts}$ on the *Interval*. For this purpose, we consider 10 base instances and solve them using CPLEX. After that, we fix the traveling constraints from ≥ to = and solve these 10 instances once more in order to see how optimal value of *Interval* will change. As can be seen from Figure 5.10, in once case we will observe an increase in optimal value of *Interval*—in instance number 4 where *Interval* increases from 37 to 38.

There are two reasons that can explain these two type outcomes. We can expect an increase because by fixing traveling time on the main line, we actually reduce the model flexibility. In other words, a train (or perhaps a set of trains) must wait for longer time at stations in order to let the occupied track to be cleared. And because of constraint (5-32) the *Interval* may increase. Using the same reasoning, we can expect no increase in optimal value of *Interval* at all. In other words, the time saved by trains due to movement at maximum speed will be lost at stations to guarantee system safety. As general rule, fixing traveling time may increase optimal value of *Interval*, but this should be further studied.

Figure 5.11 displays the results of the eighth and last set of experiments that show the effect of fixing actual dwell time of a train at zero if $forbid_{ts} = 0$ on optimal value of *Interval*. In this experiment the assumption "that trains can stop at station even if they required to" is relaxed and its effect on optimal *Interval* is studied by looking at 15 different problem instances. The result show that in more than 50% of cases the optimal value of *Interval* increases which is true. In other words, this result shows that giving this allowance to trains to stop will create more passing options, and hence increase flexibility.

In summary, by allowing train types to stop (i) more passing options will be opened; (ii) flexibility of the system will be increased; and (iii) *Interval* value will be decreased which will result in an increase in rail line capacity.

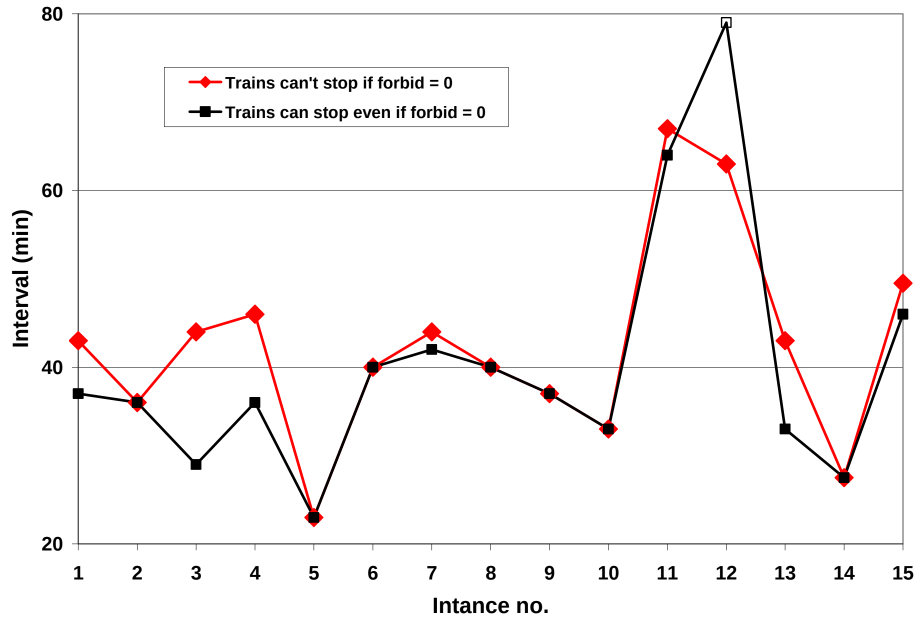


Figure 5.11 Effect of fixing actual dwell time of a train at zero if *forbid*$_{ts}$=0 value on *Interval*.

5.6. Conclusion

In this study we have presented a mixed integer linear programming model for cyclic, combined train timetabling and platforming along a bi-direction rail line. The proposed model schedules train types' arrivals and departures at stations and assigns train types to platforms in the stations so as to (i) minimize the length of the dispatching cycle – denoted by decision variable *Interval* in the model – and/or (ii) minimize total journey time of all train types. The first objective can be regarded as optimizing rail line capacity. Hundreds of randomly generated instances have been considered and solved to optimality in a reasonable amount of time using IBM ILOG CPLEX.

The experimental results yield several meaningful insights. First the ability to solve large problems instances to optimality. Second, problems of this type generally become more difficult when the number of stations, platforms, or train types increases. On the other hand, the problem becomes easier to solve when (i) both objectives are considered or (ii) train types are forbidden to stop when minimum dwell time is zero.

Chapter 6

Summary and conclusion

The first part of this research presents mixed integer programming models for cyclic train timetabling and platforming. Three models are developed and presented. The first model, presented in Chapter 3, has extended the work of Bergmann (1975) to investigate the capacity of a single track, unidirectional rail line that adheres to a cyclic timetable. In this chapter we restate the problem from Bergman (1975) using improved notation, to modify the mathematical model by adding a second objective and removing unnecessary variables, and to perform the first numerical analysis of this problem by considering hundreds of randomly generated instances with up to 70 stations. The experimental results can be summarized as follows. First, our ability to solve every problem instance to optimality in a reasonable amount of time using IBM ILOG CPLEX demonstrates the effectiveness of the model. Second, the problem becomes more difficult when (A) either of the parameter *S* or *maxExpr* are increased and everything else in unchanged; (B) parameter *hTrack* and *dMax* are simultaneously increased in such a way that neither the number of decision variables or the number of constraints (which are functions of *S* and *maxExpr* only) increases; or (C) *dMax* increases and everything else is unchanged. The case (A) is expected to happen, since increasing either *S* or *maxExpr* (or both) will increase the decision point or mathematically increase the size of the problem mathematically. Cases (B) and (C) are the direct results of expanding the solution space which requires more computation time. Third, problem difficulty appears to increase as the parameter *hTrack* increases for small values of *hTrack* but appears to decrease as *hTrack* increases for large values of *hTrack*. Fourth, the optimal value of *Interval* is smallest when the number of stations over which the same $\sum dMin_s$ value is spread is neither very small nor very large, but rather an intermediate value. Fifth, the sequence of $dMin_s$ values affects the optimal value of *Interval* in some cases but not in others. In particular, the optimal value of *Interval* seems to become more sensitive to the sequencing of the $dMin_s$ values as *hTrack* decreases or *dMax* increases. Sixth, Pareto-optimal solutions that explore the trade-off between cycle length and the local train's dwell time at all stations combined can be constructed by adjusting the objective function weights w_1 and w_2. Finally, since the problem has alternate optima, two methods are explored and described in order to find other optimal solutions. The first method is based on the weighted sum approach while the second one is based on the so called "fixed-and-cut" procedure. Future work on the problem might proceed in several directions. First, additional experiments might be conducted in order to fully analyze all of the relationships discussed in the preceding sections. Since, the problem has alternate optima; a method can be designed and developed mathematically to find all Pareto-optimal solutions in more structured way.

In Chapter 4, we present two mixed integer linear programming models of a cyclic, combined train timetabling and platforming which is the first attempt of integrating cyclic train timetabling and cyclic platforming via mixed integer linear program in the literature. These MILP models schedule train arrivals and departures at stations and assigns train types to platforms in the stations so as to minimize the length of the dispatching cycle and/or minimize the total stopping (dwell) time of all train types at all stations combined. The first objective—minimization of the length of the dispatching cycle—directly relates to rail line capacity. The current study generalizes the model presented in Chapter 3 in three ways. First, we consider any number of train types per cycle. Second, we allow stations to have more than one siding. Third, we allow trains to start or end at intermediate stations. Two real-world problems along with hundreds of randomly generated and real-world problem instances have been considered and solved to optimality in a reasonable amount of time using IBM ILOG CPLEX 11.2 and 12.4.

The experimental results yield several managerial insights. First, our ability to solve large problem instances to optimality—including an instance with 11 train types and [33] intermediate stations taken directly from the Japanese Shinkansen bullet train system timetable—demonstrates the effectiveness of the model. Second, problems of this type generally become more difficult when the number of stations, platforms, or train types increases. On the other hand, the problem becomes significantly easier to solve when (i) the second objective—minimizing total train dwell time—is disregarded or (ii) the dispatching cycle is fixed to a value with some "breathing room" and only the second objective is considered. Not surprisingly, the optimal cycle length increases monotonically as (i) the minimum required headway on the main line increases, (ii) extra train types are added to an existing rail line, or (iii) extra stations are added to the ends of an existing line. The optimal cycle length decreases monotonically as (i) extra platforms are added to the stations in an existing rail line or (ii) actual train dwell times are allowed to deviate by a greater amount from their respective minimum required dwell times in the stations where they stop. In addition, it is sometimes possible to increase the line capacity by doubling the number of train types dispatched per cycle via cloning. Finally, the optimal value of *Interval* can often be obtained without sacrificing too much in the form of increased train dwell times above the minimum required train dwell times in the stations. Future work on the problem presented in chapter 4 might proceed in several directions. For example, the modeling framework could be extended to consider trains running at different speeds along the main line. Also, heuristic methods that quickly solve very large problem instances could be developed.

In Chapter 5, we present a mixed integer linear programming model for cyclic, combined train timetabling and platforming along a bi-direction rail line. The proposed model schedules train type's arrivals and departures at stations and assigns train types to platforms in the stations so as to (i) minimize the length of the dispatching cycle – denoted by decision variable *Interval* in the model – and/or (ii) minimize total journey time of all train types. The first objective can be regarded as optimizing rail line capacity. Hundreds of randomly generated instances have been considered and solved to optimality in a reasonable amount of time using IBM ILOG CPLEX.

The experimental results yield several meaningful insights. First the ability to solve large problems instances to optimality. Second, problems of this type generally become more difficult when the number of stations, platforms, or train types increases. On the other hand, the problem becomes easier to solve when (i) both objectives are considered or (ii) train types are forbidden to stop when minimum dwell time is zero. Developing heuristic methods that quickly solve extremely large problem instances can be considered as future work on this problem. The model can be extended to consider the rolling stock constraints in the model and mixed integer programming model combined with heuristic can be developed to model and solve the problem.

The models developed here have other advantages and contribute indirectly to different real world problems in the railway planning process. By considering short-haul, local trains, the model can handle passenger transfer between trains. To do so, the objective function can be modified by adding another component or third objective which minimizes passenger waiting time. In another scenario, the models can be modified by fixing the primary objective at a given value, e.g. 60 minutes as it is the case in all real-world cyclic timetables, and minimize just the total traveling time and/or total travelling cost of all train types.

At the end, it should be noted that we did not assume any signaling or blocking in the math models developed in this dissertation, but by dividing each link between two stations, sidings or both we can incorporate that concept in the mixed integer program. In all models presented we do not model passenger demand fluctuation during a day—i.e. pick hour in the morning and afternoon and off-pick hour in the middle of day— but the optimal solution of the mixed integer programs can be easily modified and some trains can be removed from timetable.

References for Part I

1. Abril, M., Barber, F., Ingolotti, L., Salido, M. A., Tormos, P., & Lova, A. (2008a). An assessment of railway capacity. *Transportation Research Part E*, 44, 774–806.

2. Abril, M., Salido, M. A., & Barber, F. (2008b). Distributed search in railway scheduling problems. *Engineering Applications of Artificial Intelligence*, 21, 744–755.

3. Acuna-Agost, R. (2009). *Mathematical modeling and methods for rescheduling trains under disrupted operations*. PhD Thesis: University of Avigon.

4. Ahuja, R. K., Cunha, C. B., & Sahin, G. (2005). Network models in railroad planning and scheduling. *TutORials in operations research*, 1, 54-101.

5. Ahuja, R. K., Jha, K. C., & Liu, J. (2007). Solving real-life railroad blocking problems. *Interfaces*, 37, 404–419.

6. Albalate, D., & Bel, G. (2012). High-speed rail: lessons for policy makers from experience abroad. *Public Administration Review*, 72(3), 336-349.

7. Association of American Railroads. (2007). www. aar.org.

8. Ballas, E., & Jeroslow, R. (1972). Canonical cuts on the unit hypercube. *SIAM Journal of Applied Mathematics*, 23(1), 61-69.

9. Barnhart, C., Jin, H., & Vance, P. H. (2000). Railroad blocking: a network design application. *Operations Research*, 48 (4), 603-614.

10. Bergmann, D. R. (1975). Integer programming formulation for deriving minimum dispatch intervals on a guideway accommodating through and local public transportation services. *Transportation Planning and Technology*, 3, 27–30.

11. Billionnet, A. (2003). Using integer programming to solve the train-platforming problem. *Transportation Science*, 37, 213-222.

12. Brännlund, U., Lindberg, P. O., Nõu, A., & Nilsson, J.-E. (1998). Railway timetabling using Lagrangian relaxation. *Transportation Science*, 32, 358–369.

13. Burdett, R. L., & Kozan, E. (2006). Techniques for absolute capacity determination in railways. *Transportation Research Part B*, 40, 616–632.

14. Burdett, R. L., & Kozan, E. (2009a). Techniques for inserting additional trains into existing timetables. *Transportation Research Part B*, 43, 821–836.

15. Burdett, R. L., & Kozan, E. (2009b). Techniques for restricting multiple overtaking conflicts and performing compound moves when constructing new train schedules. *Mathematical and Computer Modelling*, 50, 314–328.

16. Burdett, R. L., & Kozan, E. (2009c). Scheduling trains on parallel lines with crossover points. *Journal of Intelligent Transportation Systems*, 13(4), 171-187.

17. Burdett, R. L., & Kozan, E. (2010a). A disjunctive graph model and framework for constructing new train schedules. *European Journal of Operational Research*, 200, 85–98.

18. Burdett, R. L., & Kozan, E. (2010b). A sequencing approach for creating new train timetables. *OR Spectrum*, 32, 163–193.

19. Bussieck, M. (1998). *Optimal Lines in Public Rail Transport*. Ph.D. thesis: TU Braunschweig.

20. Bussieck, M. R., Kreuzer, P., & Zimmermann, U. T. (1996). Optimal lines for railway systems. *European Journal of Operational Research*, 96, 54-63.

21. Bussieck, M. R., Winter, T., & Zimmermann, U. T. (1997). Discrete optimization in public rail transport. *Mathematical Programming*, 79, 415–444.

22. Cacchiani, V., Caprara, A., & Toth, P. (2010). Non-cyclic train timetabling and comparability graphs. *Operations Research Letters*, 38, 179–184.

23. Cacchiani, V., Caprara, A., & Fischetti, M. (2012). A Lagrangian heuristic for robustness, with an application to train timetabling. *Transportation Science*, *46*(1), 124-133.

24. Cacchiani, V., Galli, L., & Toth, P. (2014a). A tutorial on non-periodic train timetabling and platforming problems. Euro Journal on Transportation and logistics, 1-36, doi: 10.1007/s13676-014-0046-4.

25. Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., & Wagenaar J. (2014b). An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B*, 63, 15-37.

26. Cadarso, L., & Marín, Á. (2012). Integration of timetable planning and rolling stock in rapid transit networks. *Annals of Operations Research*, 199(1), 113-135.

27. Cai, X., & Goh, C. J. (1994). A fast heuristic for the train scheduling problem. *Computers & Operations Research*, 21 (5), 499-510.

28. Caimi, G., Burkolter, D., Herrmann, T., Chudak, F., & Laumanns, M. (2009). Design of a railway scheduling model for dense services. *Networks & Spatial Economics*, 9, 25-46.

29. Campus, J., & de Rus, G. (2009). Some stylized facts about high-speed rail: a review of HSR experiences around the world. *Transport Policy*, 16(1), 19-28.

30. Caprara, A., Fischetti, M., Guida, P., Monaci, M., Sacco, G., & Toth, P. (2001). Solution of real-world train timetabling problems. *In System Science, 2001. Proceeding of the 34th Annual Hawaii International Conference on* (pp. 10-pp). IEEE.

31. Caprara, A., Fischetti, M., Guida, P., Toth, P., & Vigo, D. (1999). Solutions of large-scale railway crew planning problems: the Italian experience. In *Computer-Aided Transit Scheduling* (pp. 1-18). Springer Berlin Heidelberg.

32. Caprara, A., Fischetti, M., & Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations Research*, 50, 851–861.

33. Caprara, A., Galli, L., & Toth, P. (2011). Solution of the train platforming problem. *Transportation Science*, 45, 246-257.

34. Caprara, A., Kroon, L., Monaci, M., Peeters, M., & Toth, P. (2007). Passenger railway optimization. In C. Barnhart & G. Laporte (Eds.). *Handbook in Operations Research and Management Science* (Vol. 14, pp. 129–187). Amsterdam: Elsevier.

35. Caprara, A., Monaci, M., Toth, P., & Guida, P. L. (2006). A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics*, 154, 738–753.

36. Carey, M. (1994). A model and strategy for train pathing with choice of lines, platforms, and routes. *Transportation Research Part B*, 28, 333-353.

37. Carey, M. & Carville, S. (2003). Scheduling and platforming trains at busy complex stations. *Transportation Research Part A*, 37, 195-224.

38. Carey, M., & Crawford, I. (2007). Scheduling trains on a network of busy complex stations. *Transportation Research Part B*, 41, 159–178.

39. Carey, M., & Lockwood, D. (1995). A mode, algorithm and strategy for train pathing. *The Journal of the Operational Research Society*, 46, 988–1005.

40. Cascetta, E., & Coppola, P. (2012). An elastic demand schedule-based multimodal assignment model for the simulation of high speed rail (HSR) systems. *Euro Journal on Transportation and Logistics*, 1(1-2), 3-27.

41. Castillo, E., Gallego, I., Ureña, J. M., & Coronado, J. M. (2009). Timetabling optimization of a single railway track line with sensitivity analysis. *TOP*, 17, 256–287.

42. Castillo, E., Gallego, I., Ureña, J. M., & Coronado, J. M. (2011). Timetabling optimization of a mixed double- and single-tracked railway network. *Applied Mathematical Modelling*, 35, 859–878.

43. Ceder, A. (1991). A procedure to adjust transit trip departure times through minimizing the maximum headway. *Computers & Operations Research*, 18, 417–431.

44. Chakroborty, P. & Vikram, D. (2008). Optimum assignment of trains to platforms under partial schedule compliance. *Transportation Research Part B*, 42, 169-184.

45. Chang, J. S., & Lee, J. H. (2008). Accessibility analysis of Korean high-speed rail: A case study of the Seoul metropolitan area. *Transport Review*, 28(1), 87-103.

46. Chertock, G., & Diaz, D. (2012). *Fundamentals of railway train control and signaling, including PTC systems*. Department of Engineering Professional Development Course Series: University of Wisconsin-Madison.

47. Chung, J. W., Oh, S. M., & Choi, I. C. (2009). A hybrid genetic algorithm for train sequencing in the Korean railway. *Omega*, 37, 555-565.

48. Claessens, M. T., van Dijk, N. M., & Zwaneveld, P. J. (1998). Cost optimal allocation of passenger lines. *European Journal of Operational Research*, 110, 474-489.

49. Cohen, J. L. (2003). *Multiobjective programming and planning*. NY, USA: Dover Publications.

50. Cordeau, J. F., Soumis, F., & Desrosiers, J. (2001). Simultaneous assignment of locomotives and cars to passenger trains. *Operations Research*, 49, 531-548.

51. Cordeau, J.-F., Toth, P., & Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science*, 32, 380–404.

52. Corman, F., D'Ariano, A., Hansen, I. A., & Pacciarelli, D. (2011). Optimal multi-class rescheduling of railway traffic. *Journal of Rail Transport Planning & Management*, 1, 14-24.

53. Cornelsen, S. & Di Stefano, G. (2007). Track assignment. *Journal of Discrete Algorithms*, 5, 250-261.

54. Crainic, T. G., Florian, M., & Leal, J.-E. (1990). A model for the strategic planning of national freight transportation by rail. *Transportation Science*, 24, 1–24.

55. Danna E., Fenelon, M., Gu , Z., & Wunderling, R. (2007). Generating multiple solutions for mixed integer programming problems. In M. Fischetti, & D. P. Williamson (Eds.), *Integer Programming and Combinatorial Optimization* (pp. 280-294). Springer.

56. D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2007). A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183, 643–657.

57. De Luca Cardillo, D., & Mione, N. (1998). *kL*-list λ coloring of graphs. *European Journal of Operational Research*, 106(1), 160-164.

58. Demange, M., Di Stefano, G., & Leroy-Beautlieu, B. (2012). On the online track assignment problem. *Discrete Applied Mathematics,* 160, 1072-1093.

59. De Oliveira, E. S. (2001). *Solving single-track railway scheduling problem using constraint programming*. Doctoral dissertation, The University of Leeds.

60. de Rus, G., & Nombela, G. (2007). Is investment in high speed rail socially profitable? *Journal of Transport Economics and Policy*, 41, 3-23.

61. Dessouky, M. M., Liu, Q., Zhao, J., & Leachman, R. C. (2006). An exact solution procedure to determine the optimal dispatching times for complex rail networks. *IIE Transactions*, 38, 141–152.

62. Dingler, M. H., Lai, Y.-C., & Barkan, C. P. L. (2009). Impact of train type heterogeneity on single-track railway capacity. *Transportation Research Record*, 2117, 41–49.

63. Dündar, S., & Şahin, İ. (2013). Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways. *Transportation Research Part C*, 27, 1-15.

64. Feigenbaum, B. (2013). High-speed rail in Europe and Asia: lessons for the United States. *Policy Study 418*. Reason Foundation.

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

66. Garey, M. R. & Johnson, D. S. (1978). "Strong" NP-completeness results: motivation, examples, and implications. *J. Assoc. Comput. Mach.*, 25, 499-508.

67. Ghoseiri, K., Szidarovszky, F., & Asgharpour, M. J. (2004). A multi-objective train scheduling model and solution. *Transportation Research Part B*, 38, 927-952.

68. Glover F., Lokketangen, A., & Woodruff, D. L. (2000). Scatter search to generate diverse MIP solutions. In M. Laguna & J. L. Gonzalez-Velarde (Eds.), *OR computing tools for modeling, optimization and simulation: interfaces in computer sciences and operations research* (pp. 299-317). Kluwer Academic Publishers.

69. Goerigk, M., & Schöbel, A. (2013). Improving the modulo simplex algorithm for large-scale periodic timetabling. *Computers & Operations Research*, *40*(5), 1363-1370.

70. Goossens, J. H. M., van Hoesel, C. P. M., & Kroon, L. G. (2004). A branch-and-cut approach for solving railway line-planning problem. *Transportation Science*, 38, 379-393.

71. Gorman, M. F. (2010). Rail research since deregulation: Past trends and evolution. *The Newsletter of the Railway Applications Section of INFORMS*, 2010, 6–7.

72. Gorman, M. F., Acharya, D., & Sellers, D. (2010). CSX railway uses OR to cash in on optimized equipment distribution. *Interfaces*, 40, 5–16.

73. Gorman, M. F., Crook, K., & Sellers, D. (2011). North American freight rail industry real-time optimized equipment distribution systems: State of the practice. *Transportation Research Part C*, **19**(1), 103-114.

74. Goverde, R. M. P. (2005). *Punctuality of railway operations and timetable stability analysis*. PhD Thesis: Delft University of Technology.

75. Goverde, R. M. P. (2007). Railway timetable stability analysis using max-plus system theory. *Transportation Research B*, 42(2), 179-201.

76. Goverde, R. M. P. (2010). A delay propagation algorithm for large-scale traffic network. *Transportation Research Part C*, 18, 269-287.

77. Guihaire, V. & Hao, J. K. (2008). Transit network design and scheduling: a global review. *Transportation Research Part A*, 42, 1251-1273.

78. Gutiérrez, J. (2001). Location, economic potential and daily accessibility: an analysis of the accessibility impact of the high-speed rail line Madrid—Barcelona—French border. *Journal of Transport Geography*, 9, 229-242.

79. Hansen, I. A., & Pachl, J. (Eds.). (2008). *Railway Timetable & Traffic*. Eurailpress.

80. Harrod, S. (2009). Capacity factors of a mixed speed railway network. *Transportation Research Part E*, 45, 830–841.

81. Harrod, S. (2011). Modeling network transition constraints with hypergraphs. *Transportation Science*, 45, 81–97.

82. Harrod, S. (2012). A tutorial on fundamental model structures for railway timetable optimization. *Surveys in Operations Research and Management Science*, 17, 85–96.

83. Harrod, S., & Schlechte, T. (2013). A direct comparison of physical block occupancy versus timed block occupancy in train timetabling formulations. *Transportation Research Part E*, 54, 50-66.

84. Heydar, M., Petering, M. E. H., & Bergmann, D. R. (2013). Mixed integer programming for minimizing the period of a cyclic railway timetable for a single track with two train types. *Computers & Industrial Engineering*, 66(1), 171-185.

85. Higgins, A., Kozan, E., & Ferreira, L. (1996). Optimal scheduling of trains on a single line track. *Transportation Research Part B*, 30, 147–161.

86. Higgins, A., Kozan, E., & Ferreira, L. (1997). Heuristic techniques for single line train scheduling. *Journal of Heuristics*, 3, 43-62.

87. Huisman, D., Kroon, L. G., Lentink, R. M., & Vromans, M. J. C. M. (2005). Operations research in passenger railway transportation. *Statistica Neerlandica*, 59, 67–497.

88. International Union of Railways. (2012). *General definitions of highspeed*.

89. Japanese high speed rail website (accessed November 11, 2012): http://english.jr-central.co.jp/info/timetable/index.html.

90. Jespersen-Groth, J., Potthoff, D., Clausen, J., Huisman, D., Kroon, L., Maroti, G., & Nielsen, M. N. (2009). *Disruption management on passenger railway transportation* (pp. 399-421). Springer Berlin Heidelberg.

91. Jovanovic, D., & Harker, P. T. (1991). Tactical scheduling of rail operations: The SCAN I system. *Transportation Science*, 25, 46–64.

92. Khisty, C. J. & Lall, B. K. (2003). *Transportation engineering: an introduction*. Upper Saddle River, NJ: Prentice Hall.

93. Khosravi, B., Bennell, J. A., & Potts, C. N. (2012). Train Scheduling and Rescheduling in the UK with a Modified Shifting Bottleneck Procedure. In D. Delling, & L. Liberti (Eds.), *ATMOS* (pp. 120-131).

94. Kraay, D., Harker, P. T., & Chen, B. (1991). Optimal pacing of trains in freight railroads: Model formulation and solution. *Operations Research*, 39, 82–99.

95. Kroon, L., Huisman, D., & Maroti, G. (2008). Optimization models for railway timetabling. In I. A. Hansen & J. Pachl (Eds.), *Railway Timetable & Traffic* (pp. 135–151). Hamburg, Germany: Eurailpress.

96. Kroon, L. G., & Peeters, L. W. P. (2003). A variable trip time model for cyclic railway timetabling. *Transportation Science*, 37, 198–212.

97. Lee, Y. & Chen, C-Y. (2009). A heuristic for the train pathing and timetabling problem. *Transportation Research Part B*, 43, 837-851.

98. Lee, S., Phalakornkule, C., Domach, M. M. & Grossmann, I. E. (2000). Recursive MILP model for finding all the alternate optima in LP models for metabolic networks. *Computers & Chemical Engineering*, 24(2-7), 711-716.

99. Liebchen, C. (2008). The first optimized railway timetable in practice. *Transportation Science*, 42, 420–435.

100. Liebchen, C., & Möhring, R. H. (2008). The modeling power of the periodic event scheduling problem: Railway timetables – and beyond. In M. Hickman, P. Mirchandani, & S. Voss (Eds.), *Computer-Aided Systems in Public Transport* (pp. 117–150). Berlin: Springer.

101. Liebchen, C., Proksch, M., & Wagner, F. H. (2008). Performance of algorithms for periodic timetable optimization. In M. Hickman, P. Mirchandani, & S. Voss (Eds.), *Computer-Aided Systems in Public Transport* (pp. 151–180). Berlin: Springer.

102. Lindner, T., & Zimmermann, U. T. (2005). Cost optimal periodic train scheduling. *Mathematical Methods of Operations Research*, 62, 281–295.

103. Liu, S. Q., & Kozan, E. (2009). Scheduling trains as a blocking parallel-machine job shop scheduling problem. *Computers & Operations Research*, 36, 2840–2852.

104. Liu, S. Q., & Kozan, E. (2011). Scheduling trains with priorities: A no-wait blocking parallel-machine job-shop scheduling model. *Transportation Science*, 45, 175–198.

105. Lusby, R. M., Larsen, J., Erghott, M., & Ryan, D. M. (2011a). Railway track allocation: Models and methods. *OR Spectrum*, 33, 843–883.

106. Lusby, R. M., Larsen, J., Erghott, M., & Ryan, D. M. (2013). A set packing inspired method for real-time junction train routing. *Computers & Operations Research*, 40(3), 713-724.

107. Lusby, R., Larsen, J., Ryan, D., & Ehrgott, M. (2011b). Routing trains through railway junctions: a new set-packing approach. *Transportation Science*, 45, 228-245.

108. Luthi, M. (2009). *Improving the Efficiency of Heavily Used Railway Networks through Integrated Real-Time Rescheduling*. PhD Dissertation, Department of Electrical Engineering and Information Technology, ETH Zurich, Switzerland.

109. Lutter, H., Pütz, T., & Spangenberg, M. (1992). Accessibility and peripherality of community regions: the role of road, long-distance railways and airport networks. *Commission of the European Communities*, Brussels.

110. Maróti, G., & Kroon, L. G. (2007). Maintenance routing for train units: The interchange Model. *Computers & Operations Research*, 34(4), 1121-1140.

111. Masson, S., & Petiot, R. (2009). Can the high speed rail reinforce tourism attractiveness? The case of the high speed rail between Perpignan (France) and Barcelona (Spain). *Technovation*, 29, 611-617.

112. Min, Y. H., Park, M. J., Hong, S. P., 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*, 45, 409-429.

113. Nachtigall, K., & Voget, S. (1996). A genetic algorithm approach to periodic railway synchronization. *Computers & Operations Research*, 23, 453–463.

114. Narayanaswami, S., & Rangaraj, N. (2012). Modelling disruptions and resolving conflicts optimally in a railway schedule. *Computers & Industrial Engineering*, 64(1), 469-481.

115. Newton, H. N., Barnhart, C., & Vance, P. H. (1998). Constructing railroad blocking plans to minimize handling costs. *Transportations Science*, 32(4), 330-345.

116. Odijk, M. A. (1996). A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research Part B*, 30, 455–464.

117. Park, Y., & Ha, H. K. (2006). Analysis of the impact of high-speed railroad service on air transport demand. *Transportation Research Part E*, 42, 95-104

118. Peeters, L., & Kroon, L. G. (2001). A cycle base optimization model for the cyclic railway timetabling problem. In S. Voss & J. R. Daduna (Eds.),

Computer-Aided Scheduling in Public Transport (pp. 275–296). Berlin: Springer.

119. Peng, Q., Zhao, J., & Wen, C. (2013). A rolling horizon-based decomposition algorithm for the railway network train timetabling problem. *International Journal of Rail Transportation*, *1*(3), 129-160.

120. Petering, M. E. H., Heydar, M., & Bergmann, D. R. (2012). Mixed integer programming for cyclic train timetabling and routing along a single track line. Submitted manuscript.

121. Peters, J. C., & Frittelli, J. (2012). Positive train control (PTC): Overview and policy issues. *Congressional Research Service*.

122. Petersen, E. R. (1974). Over-the-road transit time for a single track railway. *Transportation Science*, 8, 65–74.

123. Petersen, E. R., & Taylor, A. J. (1982). A structured model for rail line simulation and optimization. *Transportation Science*, 16, 192–206.

124. Rodriguez, J. (2007). A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B*, 41(2), 231-245.

125. Román, C., Espino, R., & Martín, J. C. (2007). Competition of high-speed train air transport: The case of Madrid—Barcelona. *Journal of Air Transport Management*, 13, 277-284.

126. Salido, M. A., & Barber, F. (2009). Mathematical solutions for solving periodic railway transportation. *Mathematical Problems in Engineering*, 2009, 19. (Article ID 728916).

127. Schlechte, T. (2012). *Railway track allocation: models and algorithms*. PhD Thesis: Technical University of Berlin.

128. Schöbel, A. (2012). Line planning in public transportation: models and methods. *OR Spectrum*, 34, 491-510.

129. Schrijver, A. (1993). Minimum circulation of railway stock. *CWI Quarterly*, 6, 205-217.

130. Serafini, P., & Ukovich, W. (1989). A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2, 550–581.

131. Shafia, M. A., Sadjadi, S. J., Jamili, A., Tavakkoli-Moghaddam, R., & Pourseyed-Aghaee, M. (2012). The periodicity and robustness in a single-track train scheduling problem. *Applied Soft Computing*, 12, 440-452.

132. Siebert, M., & Goerigk, M. (2013). An experimental comparison of periodic timetabling models. *Computers & Operations Research*, 40, 2251-2259.

133. Taiwanese high speed rail website (accessed September 15, 2012): http://www.thsrc.com.tw/en/index.asp.

134. UIC High Speed Department. (November 2013). High-Speed lines in the world.

135. Vaidyanathan, B., Jha, K. C., & Ahuja, R. K. (2007). Multicomodity network flow approach to the railroad crew-scheduling problem. *IBM Journal of Research and Development*, 51(3-4), 325-344.

136. Verma, M., Verter, V., & Gendreau, M. (2011). A tactical planning model for railroad transportation of dangerous goods. *Transportation Science*, 45, 163–174.

137. Yang, L., Gao, Z., & Li, K. (2010). Passenger train scheduling on a single-track or partially double-track railway with stochastic information. *Engineering Optimization*, 42, 1003-1022.

138. Yao, E., & Morikawa, T. (2005). A study of an integrated intercity travel demand model. *Transportation Research Part B*, 39, 367-381.

139. Zhou, X., & Zhong, M. (2005). Bicriteria train scheduling for high-speed passenger railroad planning applications. *European Journal of Operational Research*, 167, 752–771.

140. Zhou, X., & Zhong, M. (2007). Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bound. *Transportation Research Part B*, 41, 320–341.

141. Zwaneveld, P. J., Kroon, L. G., Romeijn, H. E., Salomon, M., Dauzere-Peres, S., van Hoesel, S. P. M., & Ambergen, H. W. (1996). Routing trains through railway stations: model formulation and algorithms. *Transportation Science*, 30, 181-194.

142. Zwaneveld, P. J., Kroon, L. G., & van Hoesel, S. P. M. (2001). Routing trains through a railway station based on a node packing model. *European Journal of Operational Research*, 128, 14-33.

Appendix for Part I

Table A.1 All unique optimal solutions for problem with S = 10, $hTrack$ = 10, $dMax$ = 72

Root (**1**)
$Y_{11} = 1$
$Y_{21} = 1$
$Y_{31} = 1$, $Y_{32} = 1$
$Y_{41} = 1$, $Y_{42} = 1$, $Y43 = 1$
$Y_{51} = 1$, $Y_{52} = 1$, $Y_{53} = 1$, $Y_{54} = 1$
$Y_{61} = 1$, $Y_{62} = 1$, $Y_{63} = 1$, $Y_{64} = 1$
$Y_{71} = 1$, $Y_{72} = 1$, $Y_{73} = 1$, $Y_{74} = 1$
$Y_{81} = 1$, $Y_{82} = 1$, $Y_{83} = 1$, $Y_{84} = 1$, $Y_{85} = 1$
$Y_{91} = 1$, $Y_{92} = 1$, $Y_{93} = 1$, $Y_{94} = 1$, $Y_{95} = 1$, $Y_{96} = 1$
$Y_{101} = 1$, $Y_{102} = 1$, $Y_{103} = 1$, $Y_{104} = 1$, $Y_{105} = 1$, $Y_{106} = 1$, $Y_{107} = 1$
$Y_{11} = 0$ (**2**)
$Y_{21} = 1$
$Y_{31} = 1$
$Y_{41} = 1$, $Y_{42} = 1$
$Y_{51} = 1$, $Y_{52} = 1$, $Y_{53} = 1$
$Y_{61} = 1$, $Y_{62} = 1$, $Y_{63} = 1$
$Y_{71} = 1$, $Y_{72} = 1$, $Y_{73} = 1$
$Y_{81} = 1$, $Y_{82} = 1$, $Y_{83} = 1$, $Y_{84} = 1$
$Y_{91} = 1$, $Y_{92} = 1$, $Y_{93} = 1$, $Y_{94} = 1$, $Y_{95} = 1$
$Y_{101} = 1$, $Y_{102} = 1$, $Y_{103} = 1$, $Y_{104} = 1$, $Y_{105} = 1$, $Y_{106} = 1$
$Y_{11} = 0$, $Y_{21} = 1$, $Y_{32}=1$ (**3**)
$Y_{21} = 1$
$Y_{31} = 1$, $Y_{32} = 1$
$Y_{41} = 1$, $Y_{42} = 1$, $Y_{43} = 1$
$Y_{51} = 1$, $Y_{52} = 1$, $Y_{53} = 1$, $Y_{54} = 1$
$Y_{61} = 1$, $Y_{62} = 1$, $Y_{63} = 1$, $Y_{64} = 1$
$Y_{71} = 1$, $Y_{72} = 1$, $Y_{73} = 1$, $Y_{74} = 1$
$Y_{81} = 1$, $Y_{82} = 1$, $Y_{83} = 1$, $Y_{84} = 1$, $Y_{85} = 1$
$Y_{91} = 1$, $Y_{92} = 1$, $Y_{93} = 1$, $Y_{94} = 1$, $Y_{95} = 1$, $Y_{96} = 1$
$Y_{101} = 1$, $Y_{102} = 1$, $Y_{103} = 1$, $Y_{104} = 1$, $Y_{105} = 1$, $Y_{106} = 1$, $Y_{107} = 1$
$Y_{11} = 0$, $Y_{21} = 1$, $Y_{32} = 0$, $Y_{42} = 1$, $Y_{53} = 1$, $Y_{64} = 1$ (**4**)
$Y_{21} = 1$
$Y_{31} = 1$
$Y_{41} = 1$, $Y_{42} = 1$
$Y_{51} = 1$, $Y_{52} = 1$, $Y_{53} = 1$
$Y_{61} = 1$, $Y_{62} = 1$, $Y_{63} = 1$, $Y_{64} = 1$
$Y_{71} = 1$, $Y_{72} = 1$, $Y_{73} = 1$, $Y_{74} = 1$
$Y_{81} = 1$, $Y_{82} = 1$, $Y_{83} = 1$, $Y_{84} = 1$, $Y_{85} = 1$
$Y_{91} = 1$, $Y_{92} = 1$, $Y_{93} = 1$, $Y_{94} = 1$, $Y_{95} = 1$, $Y_{96} = 1$
$Y_{101} = 1$, $Y_{102} = 1$, $Y_{103} = 1$, $Y_{104} = 1$, $Y_{105} = 1$, $Y_{106} = 1$, $Y_{107} = 1$
$Y_{11} = 0$, $Y_{21} = 1$, $Y_{32} = 0$, $Y_{42} = 1$, $Y_{53} = 1$, $Y_{64} = 0$, $Y_{74} = 1$ (**5**)
$Y_{21} = 1$
$Y_{31} = 1$
$Y_{41} = 1$, $Y_{42} = 1$
$Y_{51} = 1$, $Y_{52} = 1$, $Y_{53} = 1$
$Y_{61} = 1$, $Y_{62} = 1$, $Y_{63} = 1$
$Y_{71} = 1$, $Y_{72} = 1$, $Y_{73} = 1$, $Y_{74} = 1$
$Y_{81} = 1$, $Y_{82} = 1$, $Y_{83} = 1$, $Y_{84} = 1$, $Y_{85} = 1$
$Y_{91} = 1$, $Y_{92} = 1$, $Y_{93} = 1$, $Y_{94} = 1$, $Y_{95} = 1$, $Y_{96} = 1$
$Y_{101} = 1$, $Y_{102} = 1$, $Y_{103} = 1$, $Y_{104} = 1$, $Y_{105} = 1$, $Y_{106} = 1$, $Y_{107} = 1$
$Y_{11} = 1$ (**6**)

$Y_{11} = 1$ $Y_{21} = 1,\ Y_{22} = 1$ $Y_{31} = 1,\ Y_{32} = 1$ $Y_{41} = 1,\ Y_{42} = 1,\ Y_{43} = 1$ $Y_{51} = 1,\ Y_{52} = 1,\ Y_{53} = 1,\ Y_{54} = 1$ $Y_{61} = 1,\ Y_{62} = 1,\ Y_{63} = 1,\ Y_{64} = 1$ $Y_{71} = 1,\ Y_{72} = 1,\ Y_{73} = 1,\ Y_{74} = 1$ $Y_{81} = 1,\ Y_{82} = 1,\ Y_{83} = 1,\ Y_{84} = 1,\ Y_{85} = 1$ $Y_{91} = 1,\ Y_{92} = 1,\ Y_{93} = 1,\ Y_{94} = 1,\ Y_{95} = 1,\ Y_{96} = 1$ $Y_{101} = 1,\ Y_{102} = 1,\ Y_{103} = 1,\ Y_{104} = 1,\ Y_{105} = 1,\ Y_{106} = 1,\ Y_{107} = 1$
$Y_{11} = 1,\ Y_{22} = 1,\ Y_{33} = 0,\ Y_{43} = 1,\ Y_{54} = 1,\ Y_{65}=1$ (**7**)
$Y_{11} = 1$ $Y_{21} = 1,\ Y_{22} = 1$ $Y_{31} = 1,\ Y_{32} = 1$ $Y_{41} = 1,\ Y_{42} = 1,\ Y_{43} = 1$ $Y_{51} = 1,\ Y_{52} = 1,\ Y_{53} = 1,\ Y_{54} = 1$ $Y_{61} = 1,\ Y_{62} = 1,\ Y_{63} = 1,\ Y_{64} = 1,\ Y_{65} = 1$ $Y_{71} = 1,\ Y_{72} = 1,\ Y_{73} = 1,\ Y_{74} = 1,\ Y_{75} = 1$ $Y_{81} = 1,\ Y_{82} = 1,\ Y_{83} = 1,\ Y_{84} = 1,\ Y_{85} = 1,\ Y_{86} = 1$ $Y_{91} = 1,\ Y_{92} = 1,\ Y_{93} = 1,\ Y_{94} = 1,\ Y_{95} = 1,\ Y_{96} = 1\ Y_{97} = 1$ $Y_{101} = 1,\ Y_{102} = 1,\ Y_{103} = 1,\ Y_{104} = 1,\ Y_{105} = 1,\ Y_{106} = 1,\ Y_{107} = 1,\ Y_{108} = 1$
$Y_{11} = 1,\ Y_{22} = 1,\ Y_{33} = 0,\ Y_{43} = 1,\ Y_{54} = 1,\ Y_{65} = 0,\ Y_{75} = 1$ (**8**)
$Y_{11} = 1$ $Y_{21} = 1,\ Y_{22} = 1$ $Y_{31} = 1,\ Y_{32} = 1$ $Y_{41} = 1,\ Y_{42} = 1,\ Y_{43} = 1$ $Y_{51} = 1,\ Y_{52} = 1,\ Y_{53} = 1,\ Y_{54} = 1$ $Y_{61} = 1,\ Y_{62} = 1,\ Y_{63} = 1,\ Y_{64} = 1$ $Y_{71} = 1,\ Y_{72} = 1,\ Y_{73} = 1,\ Y_{74} = 1,\ Y_{75} = 1$ $Y_{81} = 1,\ Y_{82} = 1,\ Y_{83} = 1,\ Y_{84} = 1,\ Y_{85} = 1,\ Y_{86} = 1$ $Y_{91} = 1,\ Y_{92} = 1,\ Y_{93} = 1,\ Y_{94} = 1,\ Y_{95} = 1,\ Y_{96} = 1\ Y_{97} = 1$ $Y_{101} = 1,\ Y_{102} = 1,\ Y_{103} = 1,\ Y_{104} = 1,\ Y_{105} = 1,\ Y_{106} = 1,\ Y_{107} = 1,\ Y_{108} = 1$
$Y_{11} = 1,\ Y_{22} = 0,\ Y_{32} = 1,\ Y_{43} = 1$ (**9**)
$Y_{11} = 1$ $Y_{21} = 1$ $Y_{31} = 1,\ Y_{32} = 1$ $Y_{41} = 1,\ Y_{42} = 1,\ Y_{43} = 1$ $Y_{51} = 1,\ Y_{52} = 1,\ Y_{53} = 1,\ Y_{54} = 1$ $Y_{61} = 1,\ Y_{62} = 1,\ Y_{63} = 1,\ Y_{64} = 1,\ Y_{65} = 1$ $Y_{71} = 1,\ Y_{72} = 1,\ Y_{73} = 1,\ Y_{74} = 1,\ Y_{75} = 1$ $Y_{81} = 1,\ Y_{82} = 1,\ Y_{83} = 1,\ Y_{84} = 1,\ Y_{85} = 1,\ Y_{86} = 1$ $Y_{91} = 1,\ Y_{92} = 1,\ Y_{93} = 1,\ Y_{94} = 1,\ Y_{95} = 1,\ Y_{96} = 1\ Y_{97} = 1$ $Y_{101} = 1,\ Y_{102} = 1,\ Y_{103} = 1,\ Y_{104} = 1,\ Y_{105} = 1,\ Y_{106} = 1,\ Y_{107} = 1,\ Y_{108} = 1$
$Y_{11} = 1,\ Y_{22} = 0,\ Y_{32} = 1,\ Y_{43} = 1,\ Y_{65} = 0$ (10)
$Y_{11} = 1$ $Y_{21} = 1$ $Y_{31} = 1,\ Y_{32} = 1$ $Y_{41} = 1,\ Y_{42} = 1,\ Y_{43} = 1$ $Y_{51} = 1,\ Y_{52} = 1,\ Y_{53} = 1,\ Y_{54} = 1$ $Y_{61} = 1,\ Y_{62} = 1,\ Y_{63} = 1,\ Y_{64} = 1$ $Y_{71} = 1,\ Y_{72} = 1,\ Y_{73} = 1,\ Y_{74} = 1,\ Y_{75} = 1$ $Y_{81} = 1,\ Y_{82} = 1,\ Y_{83} = 1,\ Y_{84} = 1,\ Y_{85} = 1,\ Y_{86} = 1$ $Y_{91} = 1,\ Y_{92} = 1,\ Y_{93} = 1,\ Y_{94} = 1,\ Y_{95} = 1,\ Y_{96} = 1\ Y_{97} = 1$ $Y_{101} = 1,\ Y_{102} = 1,\ Y_{103} = 1,\ Y_{104} = 1,\ Y_{105} = 1,\ Y_{106} = 1,\ Y_{107} = 1,\ Y_{108} = 1$

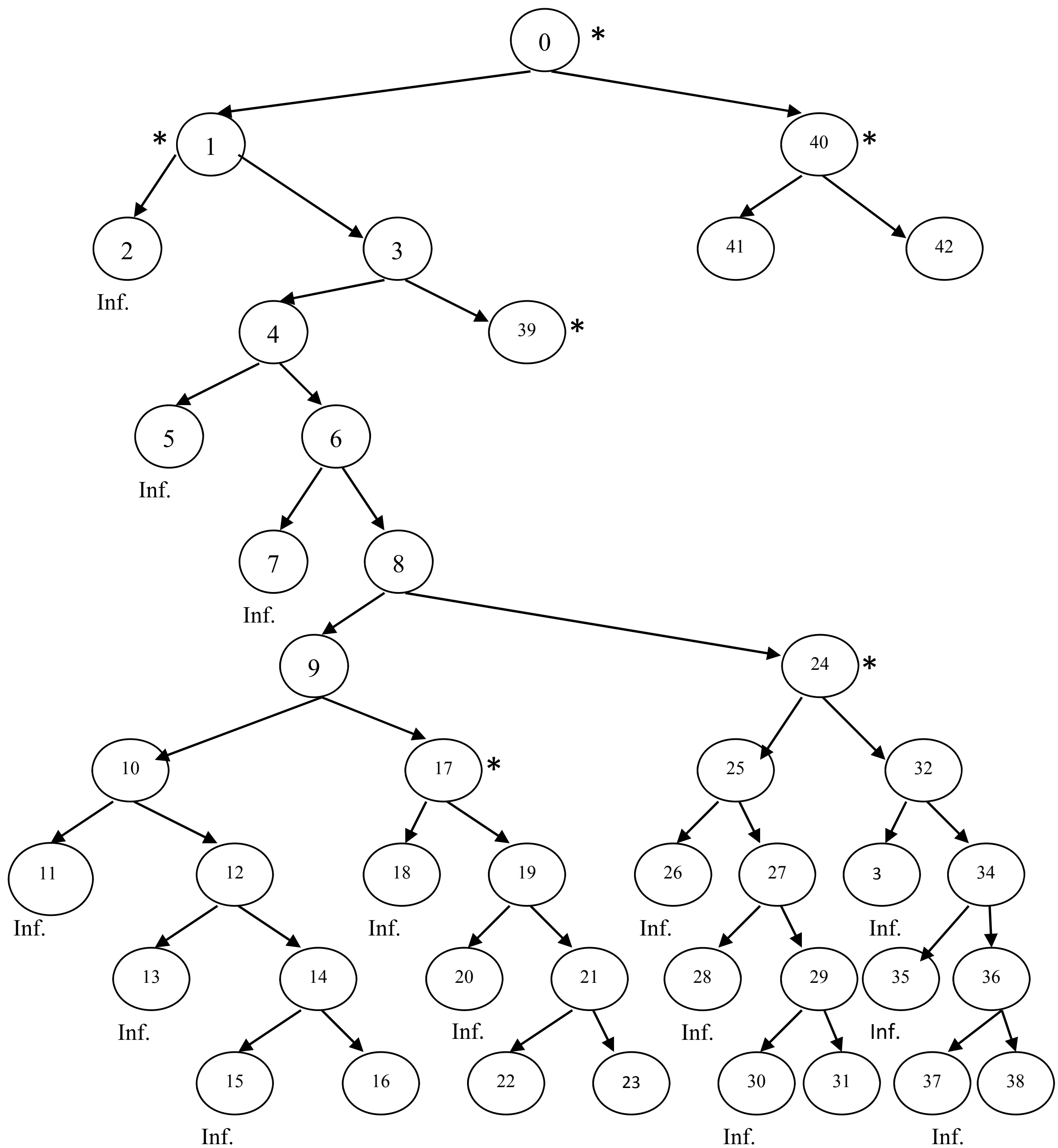


Figure A.1 Tree for counting the number of optimal solutions

Part II

A Bus Routing Problem with Application in Transit-Based Emergency Evacuation

Chapter 7

Introduction and literature review

7.1. Introduction

Natural and man-made disasters (e.g., hurricanes, floods, terrorist attacks and chemical spills) could cause huge economic loss and damage to society. In some cases like hurricanes there might be some information about the time and location. Therefore, communities can plan and relocate threatened or affected populations to a safer area. These planning activities are commonly referred as emergency evacuation. During the evacuation process, people would usually use their own vehicles to evacuate from the affected zones. However, there are some cases where people may not have access to reliable personal vehicles or using personal cars is not possible; then they need to rely on other forms of transportation. Different modes of transportation can be used to evacuate people, such as public transit, school buses, charter buses, demand-responsive vans, rail, and ambulances. According to Sayyady and Eksioglu (2010) public transit plays a crucial role in all phases of emergency management: mitigation, preparedness, response and recovery. These processes involve various complexities and are often challenged by the limited capacity of transportation system. This part of dissertation focuses on the role of public transit for the preparedness phase.

The second part of this dissertation presents a mathematical model along with a solution algorithm to design a plan for evacuating the car-less (also referred to as transit-dependent) using public transit to a safer area. The mathematical model combines two networks—pedestrian and vehicular—where the former assigns evacuees from parking lots to pick-up points considering the shortest distance and pickup point capacity, and the latter designs the routes for transit vehicles (in this paper it is assumed that the only available transit vehicle type is bus, so these two terms may be used interchangeably) to transport evacuees from pick-up points to the safer area. The objective is to minimize the total time it takes to evacuate all evacuees from parking lots to the safer area. The inputs of the model are the demand and location of parking lots, capacities and locations of the pickup points, number and locations of depots, fleet size, capacity and location of the safer area. The outputs are the number of evacuees at each pickup point, the routes of the transit vehicles, and the number of evacuees assigned to each bus during the evacuation process. Since this problem belongs to class of NP-hard problems, the larger size of this model is solved using a simulated annealing algorithm combined with an exact method.

Note that the problem studied here is static, in the way that stable tables of evacuee demand and the number of buses during an evacuation period are given. The assignment of evacuees and routing of buses also use a static representation of the network condition. The importance of the problem at hand is that the proposed model can be used by disaster management authorities at the planning stage or in some scenarios at the operational stage. The model can be used to determine how the carless people should be assigned to pick-up points and identify the number of vehicles as well as vehicle routes during the evacuation process.

7.2. Review of Literature

In a review of the literature, early efforts tackling evacuation modeling primarily focused on the network flow optimization problems that are designed to optimize various types of measurement of effectiveness (MOE), such as the evacuation time, the network clearance time, the maximum network flow, the total distance by evacuees, and the shortest paths, depending on the encountered emergency situations and management requirements (Sheffi et al., 1982; Southworth, 1991; Hobeika et al., 1994; Pidd et al., 1996; Urbina and Wolshon, 2003). To represent the evolution of an evacuation process over time, a pioneering work by Chalmet et al. (1982) develop a time-space network flow model with the objective of minimizing the total clearance time, known as the quickest flow problem. Choi et al. (1988) formulate three dynamic network flow problems (i.e., maximum flow, minimum cost and quickest flow problems) for evacuation and introduced additional constraints to define link capacity as a function of the incoming flow rate. Cova and Johnson (2003) propose an innovative lane-based network routing strategy, which provides an effective approach to reduce traffic delays at intersections. Miller-Hooks and Patterson (2004) propose the time-dependent quickest flow problem in time varying capacitated evacuation networks, where link travel times and capacities vary with time. Barbarosoglu and Arda (2004) propose a two-stage stochastic programming for transportation planning based on multi-commodity, multi-modal network flow formulation in disaster response. Liu et al. (2008) develop a two-level integrated optimization model to yield evacuation routing and traffic management plans concurrently. Kalafatas and Peeta (2009) and Xie and Turnquist (2011) further expand the capacity of the evacuation network by combining the crossing-conflict elimination and the contra flow design, which have been shown to be efficient strategies in terms of better using the network capacity for evacuation (Urbanik, 2000; Kim et al., 2008; Xie et al., 2010). Lim et al. (2012) model and solved the short notice evacuation problem by capacitated network flow problem. The model finds evacuation paths as well as flows and schedules in order to maximize the total number of evacuees for short notice planning.

Although a significant contribution has been made in evacuation modeling, only a limited number of quantitative studies discuss the use of transit to evacuate people during emergency management. One stream of researchers has employed simulation-based tools to study the feasibility and performance of transit evacuation plans. Liu et al. (2007, 2008) present an integrated system that embeds the evacuation of carless people; however, the transit demand is converted into passenger car traffic in their system. Elmitiny et al. (2007) perform a simulation study to evaluate alternative plans for the deployment of transit during an emergency situation in a transit facility such as a bus depot. Evacuation strategies evaluated include traffic diversion, bus signal optimization, access restriction, different destinations, and pedestrian evacuation. Naghawi and Wolshon (2011a, 2011b) conduct a simulation-based assessment of the performance of multi-modal evacuation traffic networks. The simulation results show that buses are able to increase the total number of people evacuated from the threatened area while adding average queue length on some interstate freeway segments. Mastrogiannidou et al. (2009) develop an effective integration of the micro-simulation software package (VISTA) with transit-based emergency evacuation models. A heuristic was developed to assign vehicle(s) to pick-up points based on the shortest time criterion. They also studied the impact of different numbers of available buses on routing strategies.

Another group of researchers has developed mathematical optimization models to obtain the best transit evacuation strategy. Perkins et al. (2001) discuss the use of buses to evacuate people (elderly and disabled) under a no-notice scenario. They assume that buses are at a garage and optimize the departure time of buses from the garage to pick-up points to minimize the total travel time of buses. However, the routing strategy is static in their model and each bus would travel on a pre-set route to leave the affected area. The number of evacuees for each pickup point is not mentioned. Sayyady (2007) formulate the carless evacuation problem with a minimum cost flow model under additional side constraints. The model assumes that bus stops are the pick-up locations and the carless are guided to the stop closest to their current location waiting for pickup during an emergency. A Tabu search technique was used to identify evacuation routes for buses. In those studies, buses would only carry out one single trip and will not return to pick up the carless after leaving the affected area. Another study (Sayyady and Eksioglu, 2010) also develop mixed-integer linear programming models to find the optimal evacuation routes for transit.

Margulis et al. (2006) develop a binary integer-programming model to determine the assignment of buses to pick-up points and to shelters during an evacuation. The objective of their model is to maximize the amount of evacuee throughput in a given time period. However, their model assumes buses are at the pickup points at the beginning of the evacuation and regulates each bus to return to the same evacuation site. He et al. (2009) develop a stochastic optimization model to generate evacuation plans for transit-dependent residents in the event of a natural disaster. Their formulation features a location-routing problem (LRP) framework and solves for the number of shelters, their locations, the number of buses required, and their routes with the objective to minimize the total evacuation time. Comparative studies were performed to analyze single-stage and two-stage transit evacuation strategies. However, their assumption that all buses are at shelters might not be appropriate. Chen and Chou (2009) develop an optimal waiting spots and service locations selection model for transit-based emergency evacuation planning and study the impact of transit-based evacuation on a highly dense populated area based on effectiveness measures such as network clearance time, move time, delay time, total travel time, and average traffic speed.

A very recent study by Chan (2010) propose a two-stage model for carless evacuation including a location problem that aims at congregating the carless at specific locations and a routing problem with the objective to pick up the carless from these evacuation sites and deliver them to safe locations. The model explicitly considers the dynamic demand pattern of evacuees to pick-up points as well as multiple trips of buses from pickup points to shelters. However, the model does not discuss how to optimally guide evacuees to pick up points to better use the available buses

Despite the significant contribution of previous studies in transit-based evacuation, none of those studies has seamlessly integrated the interactive processes of evacuee guidance (from buildings or parking lots to pick-up points) and bus routing (from pick-up points to shelters). Such integration will combine two networks: one for pedestrians and one for fleet of buses, thus making the problem an extension of the Vehicle Routing Problems (VRP). Some related previous works on VRP are reviewed below.

VRP falls into different categories. One type of VRP is the vehicle routing problem with pickup and delivery, in which vehicles transport goods and materials (also known as loads) from origins to destinations without transhipment at intermediate facilities (Savelsbergh and Sol, 1995). Gribkovskaia et al. (2007, 2008) present a mixed integer linear programming model and a Tabu search method to solve the single vehicle routing problem with pickups and deliveries. Mitra (2007) considers the problem with split deliveries and pickups and propose a parallel clustering technique to solve it. Such a situation happens in cases when the demand exceeds vehicle capacity and existing pickup-delivery algorithms cannot be used. Gulczynski et al. (2010) consider a special case of split delivery VRP in which some customers want their service in one visit. Nagy and Salhi (2005) investigate a problem with multiple depots and developed a heuristic algorithm to solve it. In their model, they relax one common VRP assumption that every pickup must be scheduled after all deliveries. In addition to meta-heuristic algorithms, some researchers also develop exact solution techniques for pickup and delivery problems. Jin et al. (2008) propose a column generation approach for the split delivery problem.

Another category of VRP is the one with multiple depots, since the original VRP assumes that there is only one depot from which vehicles start their trip and return to when they finish their services (Cordeau et al., 1997). Extension has also been made for VRP in a dynamic environment. Haghani and Jung (2005) study the dynamic VRP with time-dependent travel times combined with multiple vehicles with different capacities. Mingyong and Erbao (2010) develop a heuristic for solving the pickup and delivery VRP with time windows. Their proposed approach improves an initial solution using the Improved Differential Evolution (IDE), and uses a penalty function to prevent generating infeasible solutions.

A literature review shows very limited efforts to extend the VRP in the context of transit-based evacuation. Most studies have neglected the integrated operation of pedestrian assignment and transit routing, which can significantly improve the performance of the transit routing in response to the evacuee demand variation and maximize the use of the available number of buses by adjusting the demand distribution of evacuees at pickup points. In response to such critical research and operational needs, this study contributes to the literature of evacuation planning in different ways: (i) it is the first study that proposes an integrated mixed integer linear program (MILP) model for a two-level evacuation problem; (ii) it is the first study that model vehicular and pedestrian networks into a single network; (iii) the first study that models the integrated network as multiple depots bus routing problem with split pickup and delivery; (iv) it is capable of seamlessly and simultaneously coordinating the evacuee guidance and transit routing process.

Chapter 8

A mixed integer programming model and algorithm for strategic evacuation planning with pedestrian guidance and bus routing

8.1. Problem description

In this chapter, we describe the two-level framework for transit-based evacuation planning and propose a mixed-integer programming formulation. The "two-level" means that two different decision processes that have been made separately in previous studies are now combined in a single problem and solved simultaneously. The first level of the model guides evacuees from buildings and parking lots to designated pickup points (e.g., bus stops), and the second level of the model dispatches and routes buses from depots to pick-up points and transports evacuees to a safe place. The proposed two-level problem can be converted into a (complete) graph, as shown in Figure 8.1, in which nodes represent the parking lots, pickup points, depots, and the safer area; arcs connecting those nodes represent the road network.

In Figure 8.1, two bus routes are highlighted: the first one, shown in solid line (depot 1 – pickup 1 – pickup 2 – safer area – depot 1) and the second one, shown in dashed line (deport 2 – pickup 3 – pickup 4 – pickup 1 – safer area – depot 2). The aim of the proposed mixed integer program is to find a sub-graph that maximizes the efficiency of the evacuation plan. In this network, evacuees are assigned to pick-up points based on the capacities of pickup points and their accessing distances. Once the evacuees are assigned to the pickup points, the evacuation route for each bus will be constructed to transfer evacuees waiting at the pickup points to a safer area. Considering the nature of this two-level problem, we formulate it as a combined vehicle routing and assignment problem. As shown in Figure 8.1, one pickup point is served by two bus trips, this is another characteristic of this problem which resembles the VRP with split services (in this case split pickup). This characteristic requires more decision to be made for a bus upon its visit at a pickup point which increases the problem complexity.

Now let us formally describe the problem. Consider a network (N, E), where N and E denote the set of nodes and arcs, respectively. N is composed of four subsets of nodes: L, a set of parking lots/building nodes where demand for evacuation is initially generated and assigned to pickup points; D, a set of depot nodes where buses are initially located and dispatched from in order to pick up evacuees; P, set of pickup points, each represents a location where evacuees are waiting for buses arrival and evacuation services; H, a set of pickup nodes and the safer area (or shelter) node. There is a set of available buses, B, each has a capacity Q_b. Parking lot/building node l has demand D_l, $l \in L$ and pickup node $p \in P$, has the capacity of C_p. Each arc (i, j) has a non-negative travel time of t_{ijb}, (i, j) $\in E$. The objective is to assign evacuees from building/parking lots to pickup points and route and schedule the buses simultaneously in order to minimize the total walking and traveling times, while satisfying all evacuee demand and without violating both pickup points and buses capacities. As mentioned, split delivery service is allowed meaning that the number of evacuees at a pickup node might be greater than the capacity of a bus, thus requiring split services.

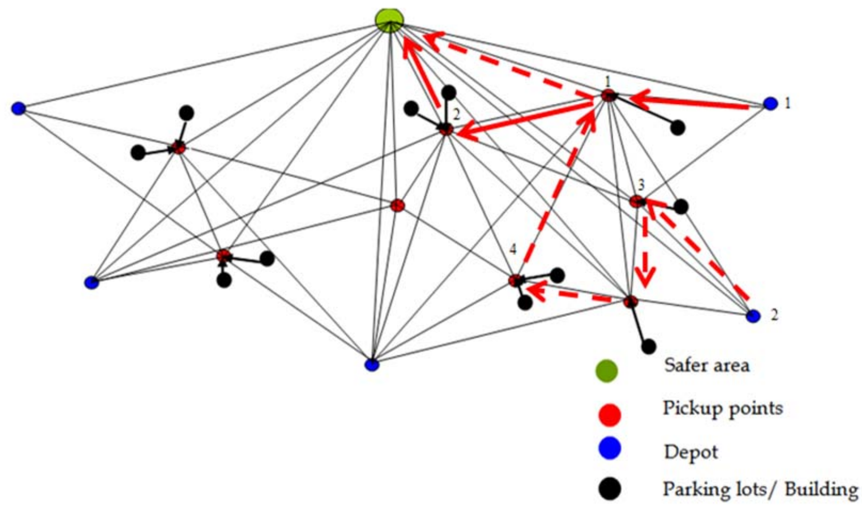


Figure 8.1 Graphical representation of the two-level evacuation problem

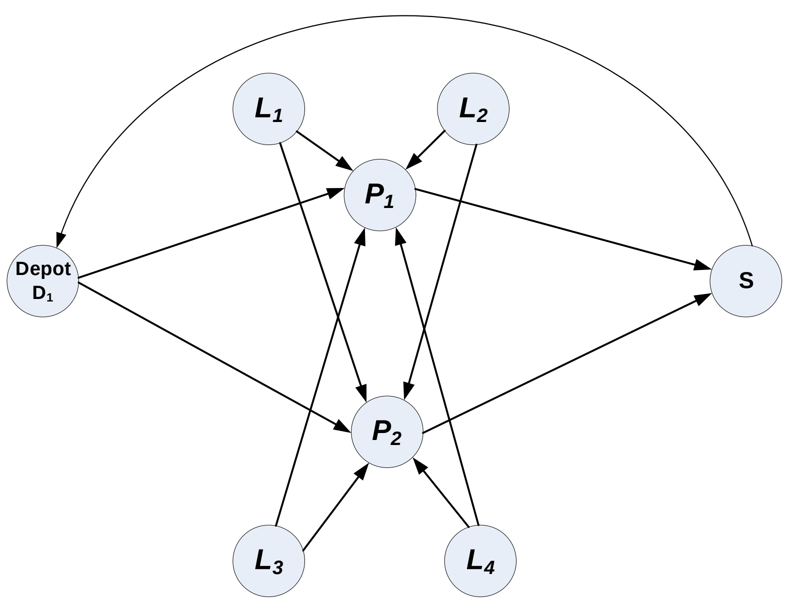


Figure 8.2 An example of evacuation network

Figure 8.2 illustrates an example of this evacuation planning network with on depot (D_1), 4 demand nodes (L_1, L_2, $L_{3,}$ and L_4), two pickup nodes (P_1 and P_2), and one shelter. This network has a different structure from the traditional routing and assignment problems, which requires a novel mixed integer program. We now list problem assumption, then present indices, parameters, decision variables and mathematical model.

To ensure that the proposed formulations are tractable and also to realistically reflect the real-world constraints, this study has employed the following assumptions in the model development: (1) There is a positive demand in the network, i.e. there is at least one evacuee waiting at parking lot; (2) Evacuees' walking time from buildings/parking lots to pick-up points and bus travel time among pickup points are given; (3) There is a super evacuation destination in the network; (4) The location and capacity of each pick-up point is given; (5) The capacity of buses are known a priori; (6) Total demand served by each bus cannot exceed its capacity; (7) Each pickup point may be served by more than one bus; and (8) Buses are restricted to return to the same deport after sending evacuees to the destination.

Table 8.1 Parameters and variables in the mathematical model

Indices	
b	Bus index $b \in B$
d	Index for depot $d \in D$
i, j, k	Vehicular node index $i, j, k \in N$
l	Parking lot / building index $l \in L$
p	Pickup point index $p \in P$
Sets	
B	Set of buses
D	Set of depots
L	Set of parking lots/building
P	Set of pick-up points
S	Set of safer areas (shelters), without loss of generality we are assuming a single shelter in this model
N	Set of nodes in vehicular networks $N = D + P + S$
H	Set of service points $H = P + S$
Parameters	
D_l	Demand at parking lot/building l; $l \in L$
C_p	Capacity of pickup point p; $p \in P$
Q_b	Capacity of bus b; $b \in B$
t_{ijb}	Average travelling time of bus b from node i to node j; $i, j \in N$, $b \in B$
w_{lp}	Average walking time from parking lot/building l to pick-up point p; $l \in L$, $p \in P$
Decision Variables	
A_{pb}	Number of evacuees at pickup point p assigned to bus b; $b \in B$, $p \in P$
Y_{lp}	Number of evacuees that is going from parking lot/building l to pick-up point p; $l \in L$, $p \in P$
T_{ijb}	Number of evacuees on bus b going from node i to node j; $j \in N$, $b \in B$
U_{jb}	An auxiliary (integer) variable for sub-tour elimination constraint in route b; $b \in B$, $j \in H$
$X_{ijb} = \begin{cases} 1 \\ 0 \end{cases}$	If arc (i, j) is traversed by bus b Otherwise; $i, j \in N$, $b \in B$
$Z_{lp} = \begin{cases} 1 \\ 0 \end{cases}$	If parking lot/building l is assigned to pick-up point p Otherwise; $l \in L$, $p \in P$

8.2. Model formulation

To facilitate model presentation, notations used hereafter are summarized in Table 8.1. The evacuation problem can be formulated as the following mixed integer linear program (MILP):

Minimize

$$\sum_{i\in N}\sum_{\substack{j\in N\\ j\neq i}}\sum_{b\in B} t_{ijb}X_{ijb} + \sum_{l\in L}\sum_{p\in P} w_{lp}Z_{lp} \tag{8-1}$$

Subject to:

$$\sum_{b\in B}\sum_{\substack{i\in N\\ i\neq j}} X_{ijb} \geq 1 \qquad \forall j \in H \tag{8-2}$$

$$\sum_{b\in B}\sum_{\substack{j\in N\\ j\neq i}} X_{ijb} \geq 1 \qquad \forall i \in H \tag{8-3}$$

$$U_{ib} - U_{jb} + |H| \times X_{ijb} \geq |P|$$

$$\forall i, j \in H, i \neq j, \forall b \in B \tag{8-4}$$

$$\sum_{i\in N} X_{ikb} - \sum_{j\in N} X_{kjb} = 0 \qquad \forall k \in N, \forall b \in B \tag{8-5}$$

$$\sum_{i=1}^{D+P} T_{ijb} + A_{jb} = \sum_{k\in H} T_{jkb} \qquad \forall j \in P, \forall b \in B \tag{8-6}$$

$$\sum_{i\in P}\sum_{b\in B} T_{ijb} = \sum_{l\in L} D_l + \sum_{k\in D}\sum_{b\in B} T_{jkb} \qquad \forall j \in S \tag{8-7}$$

$$\sum_{p\in P} A_{pb} \leq Q_b \tag{8-8}$$

$$\forall b \in B$$

$$\sum_{i\in S} T_{ijb} - \sum_{l\in P} T_{jlb} = 0 \qquad \forall b \in B, \forall j \in D \tag{8-9}$$

$$\sum_{j\in D}\sum_{b\in B} T_{ijb} = 0 \qquad i \in S \tag{8-10}$$

$$\sum_{i\in D}\sum_{j\in S}\sum_{b\in B} X_{ijb} = 0 \tag{8-11}$$

$$T_{ijb} \leq Q_b X_{ijb} \qquad \forall i, j \in N, i \neq j, \forall b \in B \tag{8-12}$$

$$\sum_{b\in B} A_{pb} - \sum_{l\in L} Y_{lp} = 0 \qquad \forall p \in P \qquad (8\text{-}13)$$

$$Y_{lp} \leq min\left\{D_l, C_p\right\} \times Z_{lp} \qquad \forall p \in P, \forall l \in L \qquad (8\text{-}14)$$

$$\sum_{p\in P} Y_{lp} = D_l \qquad \forall l \in L \qquad (8\text{-}15)$$

$$\sum_{i\in L} Y_{ip} \leq C_p \qquad \forall p \in P \qquad (8\text{-}16)$$

In this formulation, the objective function is given by Eq. (8-1), which includes two terms: the first term is dealing with routing and the second one is related to assigning evacuees to the pickup points. The first term minimizes the total time it takes by the buses to pick up all evacuees and travel through vehicular network and to drop them off at the safe area and the second term minimizes to total walking time of assigning evacuees from building/parking lots to pick-up points.

Constraint (8-2) indicates that the number of buses leaving each node should be at least one. This constraint also guarantees that each route should start from a depot. In the same fashion, constraint (8-3) guarantees that the number of buses entering a node can be greater than one and a route should be finished at a depot. Constraint (8-4) is used for sub-tour elimination in the VRP problem and is a constraint with polynomial cardinality (Laporte, 1986; Miller, 1995). Constraints (8-5) and (8-6) are flow conservation constrains for buses and evacuees, where constraint (8-5) guarantees a bus leaves a node once visits it; while constraints (8-6) guarantees that if a node is a pickup point and visited by a bus, the number of evacuees leaving that node by the bus should be equal to the total number of evacuees coming to that node on the bus and evacuees assigned to the bus at that node. Constraints (8-7), (8-9) and (8-10) guarantee that all evacuees must be dropped off at the safer area, and the capacity of each bus at the pick-up point is constrained by (8-8). By constraint (8-11), any direct route from depots to the safer area is prohibited. Constraint (8-12) ensures that the bus capacity is not violated while moving over the link connecting two nodes. In other words, this constraint in conjunction with constraint (8-8) take the bus capacity into consideration in such a way that the former focuses on the nodes and the latter focuses on the links. Constraint (8-13) ensures that all evacuees assigned to a pick-up point must be taken by the bus (buses) which is (are) passing through. Constraint (8-14) indicates that evacuees can be assigned to a pick-up point only if that pickup point has been selected. Constraints (8-15) and (8-16) guarantee that the demand at parking lots/buildings and the capacity at each pick-up point must be satisfied, respectively.

8.3. Complexity

The proposed evacuation problem is NP-hard as it can be reduced to the classical capacitated vehicle routing problem (CVRP), which is a NP-hard problem, by setting $B = 1$, $S = \emptyset$, and removing safer area and constraint (8-13) – (8-16).

8.4. Model validation

In this section, the structure and applicability of the proposed formulation is verified by one typical example and results are discussed. Note that in the mixed integer program, since each decision variable T_{ijb} and X_{ijb} is non-negative, constraints (8-10) and (8-11) can be replaced by $T_{ijb} = 0$, for $i \in S$, $j \in D$, $b \in B$, and $X_{ijb} = 0$, for $i \in D$, $j \in S$, $b \in B$, respectively. This modification will reduce the problem complexity, as we will see later even the medium-sized instances of the model are difficult to solve. In addition, since constraint (8-9) is obtained directly from constraint (8-10), and also due to the fact that all buses are empty at the time of departure from their respective depots, therefore we can have $T_{ijb} = 0$ for $i \in D$ and $j \in P$. As a result of this simplification, we have the following constraints, instead of their original forms in the model:

$$T_{ijb} = 0 \qquad \forall i \in D, \forall j \in P, \forall b \in B \tag{8-17}$$

$$T_{ijb} = 0 \qquad \forall i \in S, \forall j \in D, \forall b \in B \tag{8-18}$$

$$X_{ijb} = 0 \qquad \forall i \in D, \forall j \in S, \forall b \in B \tag{8-19}$$

The modified mathematical model was then coded into Microsoft Visual C++ 2010. Then, ILOG Concert Technology was used to define the model into C++ and call the mixed integer linear programming solver IBM ILOG CPLEX 12.4 to solve this instance within Windows 7 environment on a desktop computer with 3.1 GHz processor and 32 GB of RAM. Data used in the example are given in Table 8.2, in which buildings, pickup points and depots are represented by initials.

Table 8.2(a) lists all problem indices that will be used to solve the example. Entries in this table are self-explanatory. Table 8.2(b) depicts distances from buildings/parking lots to each pickup point. Distances between nodes of vehicular network (pickup points, depots and safer area) are given in Table 8.2(c), in which *P*'s and *D*'s stand for pickup points and depots, respectively. Tables 8.2(d) and 8.2(e) give the number of evacuees (demands) at each building/parking lot and the capacity of each pickup point, respectively. For example, the number of evacuees waiting at the first building is 10 and the first pickup point can accommodate no more than 80 evacuees. Note that data used in the numerical example are to validate the proposed model and may not be realistic considering a real-world evacuation problem. However, the proposed model is generic and can handle real-world evacuation scenarios when the data are available.

The numerical example is solved in 1153 seconds of computer time with the optimal objective function value of 215 units of time. The assignment of evacuees from buildings/parking lots to pick-up points (the first level problem) and the bus routing plans among pickup points, depots and the safer area (the second level problem) are solved concurrently with the proposed formulation. From eight available buses only seven buses are used to transport evacuees to the safer area. It should be noted that since one term of the objective function is related to route travel time, the model indirectly minimizes the number of buses used to evacuate carless people. A graphical illustration of the numerical example results is shown in Figure 8.3. In the figure, shaded circles are buildings/parking lots from which evacuees are assigned to pick-up points. The bus routing plans that take evacuees from pick-up points to the safer area and then return to their depot are also illustrated in Figure 8.3. For instance, one route (bus) starts from depot 2 to pick-up point 2, goes to the safe area and finishes its journey by returning to its origin (which is depot 2).

Table 8.2 Data used in the numerical example

(a) Number of nodes and buses of the numerical example

Parking Lots	Pickup Points	Depots	Bus	Bus capacity
10	6	3	8	50

(b) Walking time from buildings (B) to pick-up points (P) (in minutes)

	P1	P2	P3	P4	P5	P6
L1	3	4	1	2	1	3
L2	2	1	3	1	1	1
L3	1	1	1	1	2	2
L4	1	2	2	1	2	3
L5	1	2	3	3	4	1
L6	3	4	1	2	1	3
L7	2	1	3	1	1	1
L8	1	1	1	1	2	2
L9	1	2	2	1	2	3
L10	1	2	3	3	4	1

(c) Traveling time matrix for vehicular network (in minutes)

	D1	D2	D3	P1	P2	P3	P4	P5	P6	S
D1	1000	20	30	20	10	30	20	10	15	10
D2	20	1000	40	30	10	20	30	40	15	10
D3	30	40	1000	10	20	5	15	25	35	10
P1	10	20	30	1000	40	25	35	40	45	10
P2	20	10	40	30	1000	25	35	45	5	10
P3	20	20	40	30	10	1000	30	40	15	10
P4	30	40	15	10	20	5	1000	25	35	10
P5	20	40	40	30	10	20	30	1000	15	10
P6	10	20	30	45	40	25	35	40	1000	10
S	10	10	10	10	10	10	10	10	10	1000

(d) Number of evacuees at each building (unit: # of evacuees)

Building/ Parking lot	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
Demand	10	40	36	64	14	10	40	36	64	14

(e) Capacity of each pickup point (unit: # of evacuees)

Pickup points	P1	P2	P3	P4	P5	P6
Capacity	80	60	70	60	50	80

For the first level, evacuees are assigned based on the accessibility and available capacity at designated pickup points, as shown in Table 8.3. For instance, 36 evacuees are waiting at building 3; 6 of them are assigned to pick-up point 1 and the remaining 30 are assigned to pick-up point 3. On the other hand, the capacity of pickup point 1 is 48, which takes 6 evacuees from building 3, 28 from building 4, and 14 from building 10.

For the second level, the routing plan for each bus during the evacuation process is summarized in Table 8.4. Also reported in Table 8.4 is the number of evacuees taken at each pickup point and transported to the evacuation destination by each bus. It can be observed that more than one bus has been assigned to each route depending on the number of evacuees. For example, buses 1 and 3 in this table have the same route because the number of evacuees at pickup point 6 is 80, hence bus 1 will take 50, and the remaining 30 evacuees are transported by bus 3. This is because the proposed problem structure allows multiple buses on each route which is different from the assumption of the traditional vehicle routing problem. Another notable fact is that the capacity of each bus is fully used. Since bus capacity is 50, if we look at results in Table 8.4, we can observe that buses 1- 6 are used with full capacity and bus 7 carries less than capacity because the numbers of evacuees waiting in those places are less than bus capacity.

Based on the results given in Tables 8.3 and 8.4, it is apparent that the proposed mathematical model can solve this evacuation example to optimality and both the evacuee assignment and bus routing plans generated from the model are valid. The validation of the model will give us some guidelines to design a heuristic algorithm to find a good solution more quickly, which will be discussed in more details in the next section.

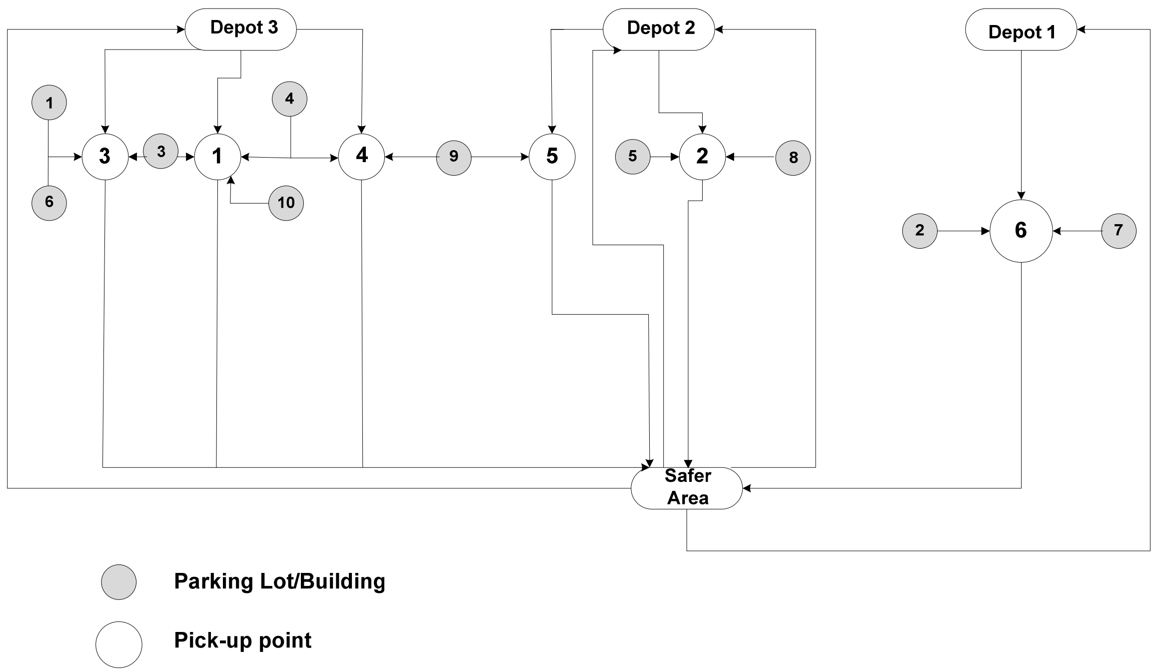


Figure 8.3 Graphical representation of the numerical example results

Table 8.3 Assignment of evacuees from buildings/parking lots to pickup points (unit: # of evacuees)

	Pick-up 1	Pick-up 2	Pick-up 3	Pick-up 4	Pick-up 5	Pick-up 6	Total
Building 1			10				10
Building 2						40	40
Building 3	6		30				36
Building 4	28			36			64
Building 5		14					14
Building 6			10				10
Building 7						40	40
Building 8		36					36
Building 9				14	50		64
Building 10	14						14
Total	48	50	50	50	50	80	

Table 8.4 The routing plan of each bus and the total of number of evacuees transported

Buses	Routing Plan	#of Evacuees/Throughput
1	Depot 2 – Pickup 6 – Safer Area – Depot 2	50
2	Depot 3 – Pickup 3 – Safer Area – Depot 3	50
3	Depot 2 – Pickup 6 – Safer Area – Depot 2	30
4	Depot 1 – Pickup 2 – Safer Area – Depot 1	50
5	Depot 1 – Pickup 5 – Safer Area – Depot 1	50
6	Depot 3 – Pickup 4 – Safer Area – Depot 3	50
7	Depot 3 – Pickup 1 – Safer Area – Depot 3	48
8	Unused.	-

8.5. Heuristic method

As discussed in Section 8.3, the proposed problem is an NP-hard problem and solving the large-scale instances is intractable. Therefore, in this section a two-stage procedure is proposed. In the first stage, a relaxed assignment problem is solved to find the evacuee demand assigned at each pickup point based on which a route for each bus is constructed through a meta-heuristic algorithm. The second stage is a simulated annealing (SA) meta-heuristic that solves the VRP sub-problem.

8.5.1. Stage I: assignment of evacuees to pick-up points

In the first stage of the algorithm, some constraints of the proposed mathematical model are relaxed in a way that only the assignment part of the model is considered. In other words, in this stage we try to assign evacuees in the buildings/parking lots to pick-up points considering demand and pick-up point capacities. This will lead to a generalized assignment model (referred as Model-I hereafter), given by:

$$\text{Minimize} \quad \sum_{p=1}^{P}\sum_{l=1}^{L} w_{lp} z_{lp} \tag{8-20}$$

$$\sum_{p\in P} Y_{lp} = D_l \qquad \forall\, l \in L \tag{8-21}$$

$$Y_{lp} \le \min\left\{D_l, C_p\right\} \times Z_{lp} \qquad \forall l \in L, \forall p \in P \tag{8-22}$$

$$\sum_{l\in L} Y_{lp} \le C_p \qquad \forall p \in P \tag{8-23}$$

The output of Model-I will provide the input for the second stage of the solution algorithm which focuses on the routing part of the model in order to transfer evacuees to the safe area. A Simulated Annealing (SA) based algorithm is employed which is detailed in the following section.

8. 5.2. Stage II: Simulated Annealing (SA) algorithm for bus routing

Simulated annealing (SA) is a probabilistic technique that has its origin in statistical mechanics first introduced by Kirkpatrick et al (1983) and used for global optimization problems. The SA generates neighboring state or solutions and accepts it, and proceeds if that state has lower energy. In case of higher energy, SA may allow the search to proceed to a neighboring state based on the value of acceptance probability. The latter case is possible to guarantee that the algorithm will not stop at local optima. Suppose that the neighboring solution selected for the next move is denoted by σ'. If this new solution does not deteriorates the objective function value, then the new solution is accepted. If the new move deteriorates the objective function value, then it is accepted with a probability $e^{(-\Delta/\theta)}$ to allow the search to escape from local optimal solution, where θ is a parameter called temperature and $\Delta = OFV(\sigma') - OFV(\sigma)$. The value of θ starts with a relatively large value, and decreases gradually to a small value close to zero. These values or parameters are controlled by a cooling scheme that specifies the initial and incremental temperature values at each stage of the algorithm.

Initial solution generation

In order to generate initial solution for the SA, we use a route constructive heuristic considering the fact that the number of evacuees waiting at pick-up points is known by solving a simple assignment problem in the Stage I. In this method it is assumed that the maximum bus capacity is equal to the maximum available capacity. Further, it is assumed that when a route is constructed, the remaining and used capacity of each bus is updated accordingly until that specific route is completed. The depot where a selected bus starts its journey from is selected randomly. The heuristic algorithm creates feasible solution since the available capacities are taken into account as the algorithm completes a route. The proposed algorithm for initial solution generation is presented as follows:

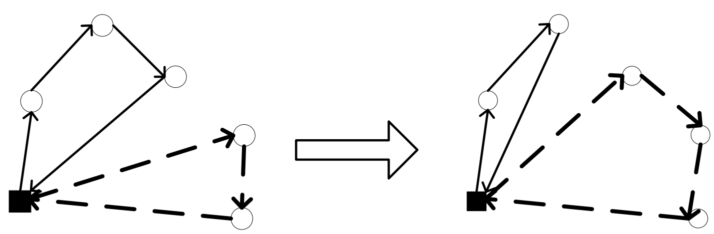
Let N be the set of nodes including depots, pick-up points and the safe area with D as the number of depots. Also let U be a set of not served nodes, this includes nodes that are not served completely, where $U \subset N - \{D+1\}$. Let B be the set of buses, and let B_0 be the set of unused buses such that $B_0 \subset B$. Also $\varepsilon_i \leq \rho_i$ is the number of customers served at node i up to now. Steps of the complete algorithm are given in Table 8.5.

Neighborhood generation mechanism

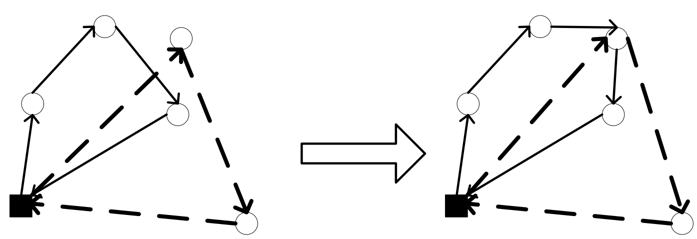
To search the solution space in order to obtain the (near) optimal solution, one has to generate neighborhood solutions. In this study an extension of 1-opt and 2-opt mechanisms are applied as proposed in the literature (Tarantilis et al., 2004). These operators are explained below and depicted in Figures 8.4 and 8.5.

- In a 1-opt operator, like the traditional operator, from the current solution two routes (each belongs to one bus) are randomly selected. A node (other than depot) is selected randomly from one route. Then, by considering the capacity of the other bus on the second route, this selected node will be added to the second route. For this operator two cases may happen. In one case, the number of evacuees waiting at a pick-up point is less than or equal to the capacity of the bus, so all evacuees can be assigned to the bus. In this operator, in order to avoid a degenerate solution, the just emptied pick-up point should be removed from the route. In the second case, the number of evacuees is greater than the capacity of the bus, therefore the number of evacuees should be split and as a result the node should be added to the second route as well. This is the split-service characteristic of the model which is similar to the split-service VRP. This operator is shown in Figure 8.4.

- In a 2-opt operator, two routes belong to two different buses are selected randomly from the current solution. Then, two nodes are chosen randomly and exchanged with each other with the observance of each bus capacity and whether or not any node(s) is (are) depot(s). This operator is shown in Figure 8.5. Table 8.5 shows the pseudo code for this initial solution generation procedure.



(a) 1-opt implementation with one depot (classical case).



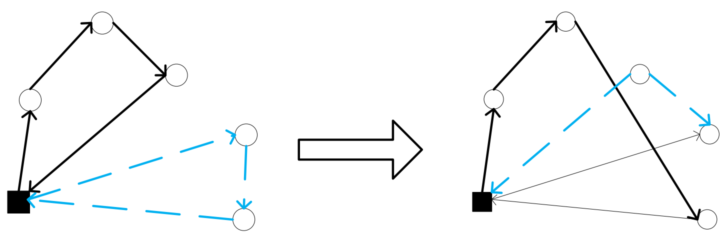
(b) 1-opt implementation with one depot (split service).

Figure8.4 1-opt implementation: (a) classical case and (b) split service

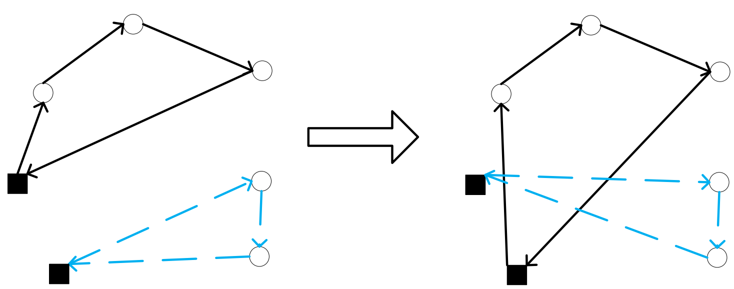
SA implementation

The SA algorithm has two loops: inside and outside. The inside loop which is the core of SA and refers to *Metropolis* procedure simulates the annealing process in the current temperature θ_r . The outside loop controls the rate of the decrease in temperature. The SA parameters are summarized in Table 8.6; and Table 8.7 depicts the pseudo code of SA meta-heuristic of the solution algorithm.

In summary, the overall solution algorithm for the proposed evacuation mixed integer linear program that combines the assignment problem and simulated annealing is presented in Figure 6 in the form of a flowchart.



(a) 2-opt for between two vehicles dispatched from one depot



(b) 2-opt for between two vehicles dispatched from two depots and depots are exchanged.

Figure 8.5 2-opt implementation

Table 8.5 Algorithm for initial solution generation

Algorithm I. Initial solution generation

Step 1

Let $U = N - \{D + 1\}$, $B_0 = B$ and $q = 0$, where q is a counter of the number of used capacity in the current allocated bus.

Step 2

If $U \neq \varnothing$ and $B_0 = \varnothing$, then the current solution is infeasible and exit; otherwise, select node i randomly where $i \in U$ and node d, $d \in D$, at random so that a bus can be assigned to start its journey.

Step 3

Dispatch bus $b \in B_0$ from depot d to node i. Let $q = q + min\{Q_b, \rho_i - \varepsilon_i\}$ and $\varepsilon_i = \varepsilon_i + min\{Q_b, \rho_i - \varepsilon_i\}$. If $\varepsilon_i = \rho_i$ then let $U = U - \{i\}$. If $q = Q_{max}$ then dispatch b to the shelter and go to step 5, else go to step 4.

Step 4

Find node j at random, where $j \in U$, and dispatch b from node i to node j. Then, let $q = q + min\{Q_b, \rho_j - \varepsilon_j\}$ and $\varepsilon_j = \varepsilon_j + min\{Q_b, \rho_j - \varepsilon_j\}$. If $\varepsilon_j = \rho_j$ then let $U = U - \{j\}$. If $q = Q_{max}$ then dispatch b to the shelter and go to step 5; otherwise let $i = j$ and repeat step 4.

Step 5

Dispatch bus b to the shelter and let $b^* = b$ and set $B_0 = B_0 - \{b^*\}$.

Step 6

Repeat steps 2 through 5 until $U \neq \varnothing$, i.e. all evacuees are transported to the shelter.

8.6. Computational results

In this section, the proposed model is solved by the heuristic developed in the previous sections and computational results are discussed and reported here.

Fifteen problem instances that can be classified into medium to large are randomly generated. The sizes of instances vary from 15-200 buildings, 6-150 pickup points, 1-3 depots, and 2-170 buses. Specifications of these instances are given in Table 8.8. Each instance can be characterized by its size which is defined by the number of parking lots, pick-up points, depots and buses, or symbolically as ***L|P|D|B***, and will be used hereafter. For example, an instance with 15 parking lots, 6 pick-up points, 3 depots and 10 buses can be represented as ***15|6|3|10***. Looking at the size of parameters will reveal that all of these instances are considered as large-scale instances for the proposed mathematical model. SA parameters are set experimentally in advance as shown in Table 8.9. The cooling scheme parameter and its effect will be discussed further in this section; therefore it will be setting up for each instance and experiment individually.

Table 8.6 Parameters of SA algorithm

MP	Number of the accepted solutions in each temperature in *Metropolis* procedure.
$MNTT$	Maximum number of consecutive temperature trails
θ_0	Initial temperature
α	Rate of decreasing temperature (cooling scheme)
σ^0, σ^n, σ^{new} and σ^{best}	Initial, current, new and the best solution or state of the optimization problem
$OFV(\sigma)$	The value of the objective function for solution σ
n	Counter for the number of the accepted solutions in each temperature
r	Counter for number of consecutive temperature trails, where θ_r is equal to temperature in iteration r.

Table 8.7 The pseudo code of SA meta-heuristic

Algorithm II. Simulated Annealing

Initialize parameters θ_0, θ_r, MP and $MNTT$.

$r = 0$, and $\sigma^{best} \leftarrow \varnothing$

Generate initial solution σ^0 by the proposed algorithm given in section 5.2.1.

$\sigma^{best} \leftarrow \sigma^0$

Do {

$n = 0$

Do {

Select an operator (1-opt or 2-opt) randomly and execute it on solution σ^n

$\sigma^{new} \xleftarrow{operator} \sigma^n$

$\Delta OFV = OFV\left(\sigma^{new}\right) - OFV\left(\sigma^n\right)$

IF $\Delta OFV < 0$ **THEN**

$\sigma^{best} \leftarrow \sigma^{new}$

$\sigma^n \leftarrow \sigma^{new}$

$n \leftarrow n+1$

ELSE

Generate $\varphi \sim U(0, 1)$ randomly and set $\vartheta = EXP\,(-\,\Delta OFV\,/\,\theta_r)$

IF $\varphi < \vartheta$ **THEN**

$\sigma^n \leftarrow \sigma^{new}$

$n \leftarrow n+1$

END IF

} while ($n < MP$ **)**

$r \leftarrow r+1$

$\theta_r \leftarrow \theta_{r-1} - \alpha\theta_{r-1}$

} while ($r < MNTT$ & $\theta_r > 0$**)**

Print σ^{best}

The generated problem instances are first solved by IBM ILOG CPLEX 12.4 and a computational time limit of 3 hours is imposed. Through this restriction, we are able to compare the performance of the algorithm in terms of the quality of the solution and running time as well as the trade-off that has to be made between these two criteria. Computation results are shown in the rightmost four columns (*Best Integer*, *Best bounds*, *Gap,* and *Time*) of Table 8.8. In this table, *Best Integer* shows the best integer value of the objective function found by CPLEX as the solution procedure proceeds. Similarly, *Best Bound* and *Gap* columns report their respective values from CPLEX. As can be seen from the table, CPLEX is able to find feasible solutions for only 5 instances within a given time limit, among which only one instance (i.e. INS01 with the optimal value of 685 unit of time) is solved to optimality. For all other instances, CPLEX is not able to construct or find the best integer values in 10800 seconds (3 hours). These results apparently show that finding the optimal solutions using the current system specifications and settings is not practically feasible. For those instance that cannot be solved to optimality (INS02 – INS05) the best integer solution have been reported in order to assess the quality of the proposed algorithm that will be discussed later.

To assess the quality of the solutions obtained by the proposed heuristic, each of the 15 problem instances was solved for 10 times with two different values for α (0.8 and 0.1), therefore a total of 300 instances are solved using the heuristic. All the best objective function values found for each trial are reported in columns 3-12 in Table 8.10 and the average objective function is shown in the column 13 of this table. Columns 14 and 15 show the best objective function values found for each problem along with the CPU time in seconds. Table 8.10 shows that the proposed methodology can handle very large instances with over 200 demand points and over 150 buses. From this table a couple of observations can be made. The first observation is the effect of cooling scheme on the quality of the solution and CPU time. As can easily be observed from Table 8.10, in all 15 instances the time required to terminate the algorithm is a (monotonically) decreasing function of α. This is true because according to the outside loop (see Table 8.7), algorithm searches more area of the solution space. Such an impact can be further illustrated by Figure 8.7, which shows the impact of parameter α on the quality of solutions obtained for a relatively small instance of 10|3|6|3. Five levels of cooling schemes are designed with α = 0.99, 0.9, 0.8, 0.5 and 0.1. As shown in Figure 8.7, the best objective function values found will improve as the value of α decreases from 0.99 to 0.1, indicating that a lower cooling rate will allow the algorithm to spend more time in searching the global optimal solution.

Table 8.8 Specifications of the 15 instances in the numerical study and results solved by CPLEX

Instance #	*L*	*P*	D	*B*	Q_b	*Best Integer*	*Best Bounds*	*Gap*	*Time(sec)*
INS01	15	6	3	10	145	685*	679	0.88%	2939
INS02	50	20	3	20	145	936	671	28.26%	10800
INS03	50	50	1	2	100	431	366	15.08%	10800
INS04	50	50	1	5	50	639	388	39.28%	10800
INS05	100	20	2	50	120	4155	1860	55.23%	10800
INS06	100	25	1	50	120	N/A			
INS07	100	25	2	50	120	N/A			
INS08	100	50	1	50	120	N/A			
INS09	100	50	2	50	120	N/A			
INS10	100	100	1	20	100	N/A			
INS11	100	100	2	20	100	N/A			
INS12	200	100	2	30	100	N/A			
INS13	200	150	2	30	100	N/A			
INS14	200	150	2	110	150	N/A			
INS15	200	150	2	170	100	N/A			

* This is the optimal value of the objective function.

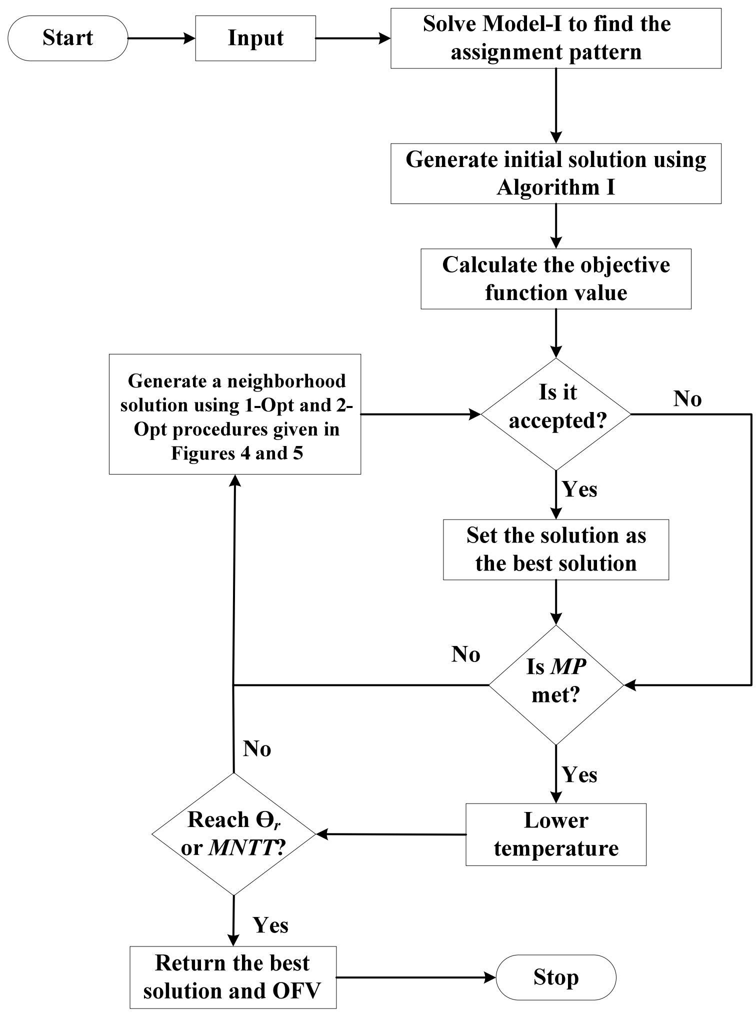


Figure 8.6 Solution flowchart for the proposed evacuation problem

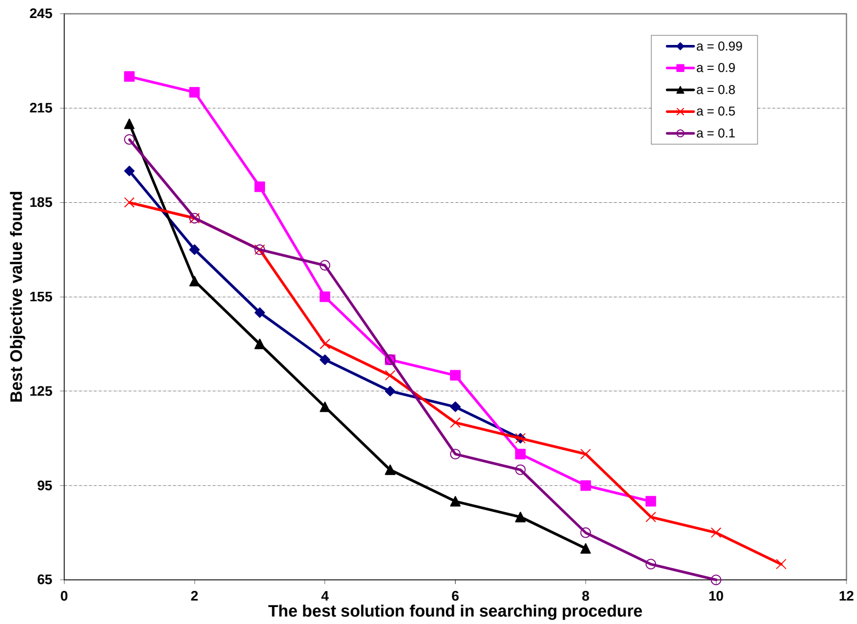


Figure 8.7 The effect of *a* on the solution quality

Table 8.9 General settings for SA parameters

MP	*MNTT*	θ_0	1-Opt rate	2-Opt rate
1000	1000	10000	0.5	0.5

The second observation is the effectiveness of the heuristic in yielding good solutions. This observation, and its immediate conclusion, can be made by comparing Tables 8.8 and 8.10. As mentioned earlier, CPLEX is able to solve only one instance to optimality, and to provide the feasible solution for four instances within 3 hours limit. From results reported in Table 8.10, we can see that the best objective value for INS01 (15|6|3|10), with α = 0.1, is 701 which is only +2.19% from optimal value reported in Table 8.8. The algorithm cannot return a good solution for INS02 (50|20|3|20). However, for INS03 - INS05 the algorithm is able to significantly improve the best objective value found by CPLEX. From Table 8.10, it can be seen that when $\alpha = 0.1$ is used, the OFV has been reduced from 431 to 409 (5.1%) and 639 to 562 (about 12.05%) for INS03 and INS04, respectively. In a similar way, for INS05 the best objective function has been reduced from 4155 to 3789 (8.81%) when $\alpha = 0.1$ was used in the algorithm. It should be noted that these instances can be classified as large scale problems and the proposed heuristic can yield a better solution than CPLEX in a much shorter amount of computational time. As mentioned earlier, we were able to find optimal (feasible) solution for only 5 instances within 3 hours of running time by CPLEX, but the proposed heuristic yields feasible and good solutions for the other 10 instances in a reasonable time window. Therefore, it can be concluded that the proposed heuristic works well in providing us with (near) optimal or feasibly good solution for medium- to large-scale problem that may arise in real transit-based evacuation situations.

To further illustrate the efficiency of the proposed heuristic, Figure 8.8 shows the convergence process of the algorithm for the instance INS04 (50|50|1|5). As shown in the figure, the algorithm was able to solve this instance in a few seconds. The results illustrate that the proposed solution algorithm performs well and converges fast to a good near optimal solution.

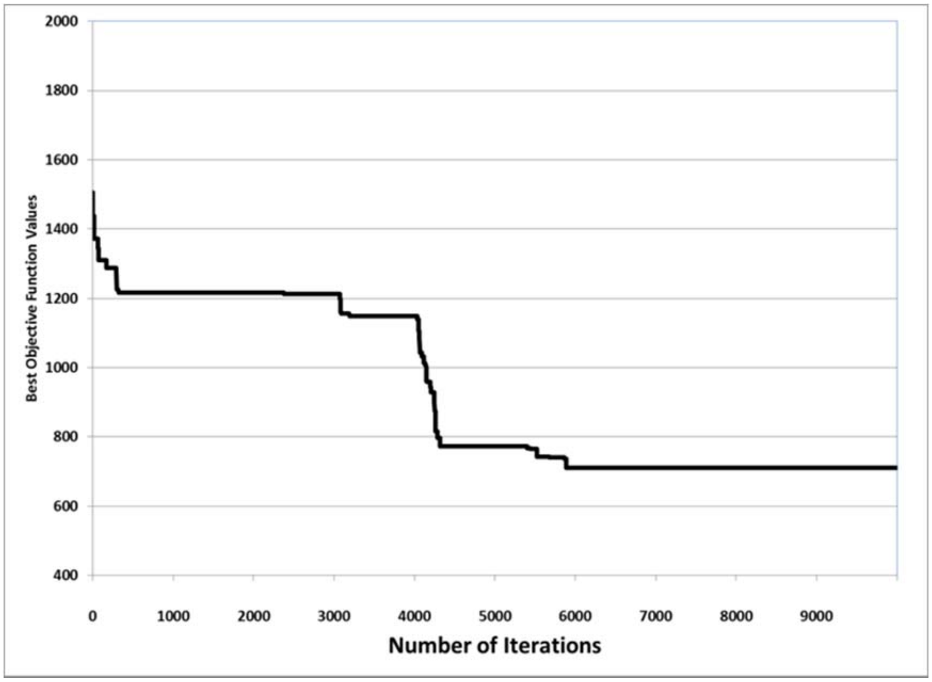


Figure 8.8 Convergence of the solution algorithm for INS04

Table 8.10 Computation results by the proposed heuristic

IS	α	Run No. 1	2	3	4	5	6	7	8	9	10	Avg. OFV	Best OFV	CPU (sec)
1	0.8	764	748	799	769	746	725	740	744	**728**	755	754.8	728	6
	0.1	719	710	719	719	730	**701**	732	710	750	748	723.8	701	640
2	0.8	3392	2589	2541	1761	**1551**	2353	3690	1560	4645	2641	2672.4	1552	11
	0.1	1586	1513	**1383**	1444	1492	14556	1456	1512	1500	1512	1497.4	1383	1270
3	0.8	719	695	647	743	**622**	526	647	732	667	711	743.9	622	5
	0.1	511	**409**	525	543	494	527	530	506	534	541	520	409	640
4	0.8	728	727	731	**687**	732	786	697	690	724	700	720.2	687	5
	0.1	593	581	622	**562**	593	563	584	588	612	573	587.1	562	620
5	0.8	4556	4640	**4396**	4601	4537	4584	4611	4630	4400	4712	4567	4396	25
	0.1	**3789**	3912	3920	4223	4062	3945	4011	4160	4004	3909	3993.5	3789	3200
6	0.8	**5028**	5097	5146	5076	5170	5190	5133	5135	5291	5201	5146.7	5028	26
	0.1	4779	4862	4853	4998	4836	4895	4769	4795	**4746**	4799	4833.2	4746	3400
7	0.8	4829	5027	5041	**4620**	4785	4859	5002	4992	5004	4857	4901.6	4620	28
	0.1	4439	4290	4348	4140	5025	4242	4364	4215	4177	**3835**	4227.5	3935	3419
8	0.8	4959	4711	**4706**	4915	4909	4987	4939	4838	4956	4826	4874.6	4706	28
	0.1	4782	5076	**4681**	4851	4797	4741	4786	4734	4752	4692	4789.2	4681	3520
9	0.8	4888	4796	4719	4858	4753	**4618**	4693	4664	4799	4735	4752.3	4618	26
	0.1	4698	4546	4602	4598	4853	4609	4621	4595	**4493**	4582	4619.7	4493	3482
10	0.8	**2439**	2444	2582	2703	2735	2683	2595	2698	2592	2455	2589.6	2439	11
	0.1	1374	1336	1250	1338	1336	1306	1311	1327	**1222**	1339	1313.9	1222	1147
11	0.8	2604	2427	**2205**	2564	2379	2424	2604	2628	2424	2234	2449.3	2205	11
	0.1	1157	1234	1298	1234	**1140**	1156	1175	1179	**1112**	1247	1193.2	1112	1142
12	0.8	3432	3349	3474	3446	3440	2293	3344	**3121**	3394	3262	3355.5	3121	19
	0.1	3594	3253	**3032**	3302	3303	3433	3130	3178	3152	3061	3243.8	3032	1848
13	0.8	**3350**	3927	3807	3557	3764	3800	3920	3714	3661	3747	3704.7	3350	20
	0.1	3253	3279	3200	3345	3306	3260	3591	3330	3328	**3227**	3321.9	3227	1898
14	0.8	11648	11393	**11065**	11519	11677	11356	11999	11242	11798	11325	11492	11065	55
	0.1	11525	11184	**10898**	11373	11447	11211	11431	10903	11564	11564	11.310	10898	5616
15	0.8	15206	14847	15125	15013	14828	14732	14763	15077	**14403**	15205	14916	14403	81
	0.1	14961	14509	14566	14633	14488	14612	14707	14805	**14273**	14603	14618	14273	6576

Chapter 9

Conclusion and future work

The second part of this dissertation presents a mixed integer linear program (MILP) for evacuation planning in highly populated urban zones where a potentially large number of pedestrians depend on transit for evacuation. The uniqueness of the proposed model lies in its capability to concurrently operate the interactive processes of evacuee guidance (from buildings or parking lots to pick-up points) and bus routings (from pickup points to shelters). Such integration will significantly improve the performance of the transit routing in response to the evacuee demand variation and maximize the use of the available number of buses by adjusting the demand distribution of evacuees at pick-up points. The feasibility and applicability of the proposed model is illustrated with an illustrative example solved to optimality. Results show that the proposed model can yield valid and detailed evacuee guiding and transit routing plans during the evacuation.

A two-stage algorithm is designed to solve this combined routing and assignment problem for large-scale instances. In the first stage, an assignment sub-problem is solved to determine the optimal assignment of evacuees to pick-up points. In the second stage, a SA-based algorithm is designed to solve the routing part of the model. In the proposed SA algorithm an extension of 1-Opt and 2-Opt operators applied to the multiple depot capacitated split service vehicle routing problem.

The proposed model and solution algorithm are validated with an extensive number of tests. Compared to solutions obtained by CPLEX, the proposed algorithm is demonstrated to be able to yield effective solutions to the proposed problem for large and realistic evacuation scenarios in a reasonable amount of time. Trade-off between the quality of solutions and the computational time can be easily made by adjusting the cooling rate parameter in the proposed algorithm, which offers the flexibility for the transportation authorities to make decisions depending on the time and budget constraints.

Note that the problem studied here is static, in a way that a stable table is given of evacuee demand and the number of buses during an evacuation period. The assignment of evacuees and routing of buses also use a static representation of the network condition. Therefore, this model is very useful at the initial stage of strategic evacuation planning. Extending the model to an explicitly dynamic setting with time-varying demand generation rates and travel times is a worthwhile direction for further work.

References for Part II

1. Barbarosoglu, G., & Arda, Y, (2004). A two-stage stochastic programming framework for transportation planning in disaster response. *Journal of the Operational Research Society*, 55, 43-53.

2. Chalmet, L., Francis, R., & Sanders, P. (1982). Network models for building evacuation. *Management Science*, 28(1), 86-105.

3. Chan, C. P. (2010). *Large scale evacuation of carless people during short-and long-notice emergency*. Unpublished PhD Thesis: The University of Arizona.

4. Chen, C., & Chou, C. (2009). Modeling and performance assessment of a transit-based evacuation plan within a contraflow simulation environment. *Transportation Research Record: Journal of the Transportation Research Board*, 2091(-1), 40-50.

5. Choi, W., Hamacher, H. W., & Tufekci, S. (1988). Modeling of building evacuation problems by network flows with side constraints. *European Journal of Operational Research*, 35(1), 98-110.

6. Cordeau, J., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2), 105-119.

7. Cova, T. J., & Johnson, J. P. (2003). A network flow model for lane-based evacuation routing. *Transportation Research A*, 37(7), 579-604.

8. Elmitiny, N., Ramasamy, S., & Radwan, E. (2007). Emergency evacuation planning and preparedness of transit facilities: Traffic simulation modeling. *Transportation Research Record: Journal of the Transportation Research Board*, 1992(1), 121-126.

9. Gribkovskaia, I., Halskau, Ø., Laporte, G., & Vlček, M. (2007). General solutions to the single vehicle routing problem with pickups and deliveries. *European Journal of Operational Research*, 180(2), 568-584.

10. Gribkovskaia, I., Laporte, G., & Shyshou, A. (2008). The single vehicle routing problem with deliveries and selective pickups. *Computers & Operations Research*, 35(9), 2908-2924.

11. Gulczynski, D., Golden, B., & Wasil, E. (2010). The split delivery vehicle routing problem with minimum delivery amounts. *Transportation Research Part E*, 46(5), 612-626.

12. Haghani, A., & Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research*, 32(11), 2959-2986.

13. He, S., Zhang, L., Song, R., Wen, Y., & Wu, D. (2009). Optimal transit routing problem for emergency evacuations. In *Transportation Research Board 88th Annual Meeting* (No. 09-0931).

14. Heydar, M., Liu, Y., Petering, M. E. H., & Yu, J. (2014). A mixed integer programming and algorithm for strategic evacuation planning with pedestrian guidance and bus routing. Submitted manuscript.

15. Hobeika, A. G., Kim, S., & Beckwith, R. E. (1994). A decision support system for developing evacuation plans around nuclear power stations. *Interfaces*, 24(5), 22-35.

16. Jin, M., Liu, K., & Eksioglu, B. (2008). A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters*, 36(2), 265-270.

17. Kalafatas, G., & Peeta, S. (2009). Planning for evacuation: Insights from an efficient network design model. *Journal of Infrastructure Systems*, 15(1), 21-30.

18. Kim, S., Shekhar, S., & Min, M. (2008). Contraflow transportation network reconfiguration for evacuation route planning. *Knowledge and Data Engineering, IEEE Transactions on*, 20(8), 1115-1129.

19. Kirkpatrick, S., Gellat, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671-680.

20. Laporte, G. (1986). Generalized subtour elimination constraints and connectivity constraints. *Journal of Operational Research Society*, 37(5), 509-514.

21. Lim, G. J., Zangeneh, S., Reza Baharnemati, M., & Assavapokee, T. (2012). A capacitated network flow optimization approach for short notice evacuation planning. *European Journal of Operational Research*, 223(1), 234-245.

22. Liu, Y., Chang, G., Lai, X., & Liu, Y. (2007, October). CAPEVACUATION: The corridor-based emergency traffic evacuation system for Washington DC. In the *14th World Congress on Intelligent Transportation Systems*.

23. Liu, Y., Chang, G., Liu, Y., & Lai, X. (2008). Corridor-based emergency evacuation system for Washington DC: System development and case study. *Transportation Research Record: Journal of the Transportation Research Board*, 2041(1), 58-67.

24. Margulis, L., Charosky, P., Fernandez, J., & Centeno, M. A. (2006, June). Hurricane evacuation decision-support model for bus dispatch. In *Fourth International Latin American and Caribbean Conference on Engineering and Technology, Puerto Rico*.

25. Mastrogiannidou, C., Boile, M., Golias, M., Theofanis, S., & Ziliaskopoulos, A. (2009). Using transit to evacuate facilities in urban areas: A micro-simulation based integrated tool. In *Proceedings of 88th Annual Meeting of the Transportation Research Board*, pp. 11-15.

26. Miller, D. L. (1995). A matching based exact algorithm for capacitated vehicle routing problems. *ORSA Journal on Computing*, 7(1), 1-9.

27. Miller-Hooks, E., & Stock Patterson, S. (2004). On solving quickest time problems in time-dependent, dynamic networks. *Journal of Mathematical Modelling and Algorithms*, 3(1), 39-71.

28. Mingyong, L., & Erbao, C. (2010). An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. *Engineering Applications of Artificial Intelligence*, 23(2), 188-195.

29. Mitra, S. (2007). A parallel clustering technique for the vehicle routing problem with split deliveries and pickups. *Journal of the Operational Research Society*, 59(11), 1532-1546.

30. Naghawi, H., & Wolshon, B. (2011a). Operation of multimodal transport systems during regional mass evacuations. In *Proceedings of 90th Annual Meeting of the Transportation Research Board*, pp. 23-27.

31. Naghawi, H., & Wolshon, B. (2011b). Performance of multi-modal evacuation traffic networks: A simulation based assessment. In *Proceedings of 90th Annual Meeting of the Transportation Research Board*, pp. 23-27.

32. Nagy, G., & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, 162(1), 126-141.

33. Perkins, J. A., Dabipi, I. K., & Han, L. D. (2001). *Modeling Transit Issues Unique to Hurricane Evacuations: North Carolina's Small Urban and Rural Areas*.

34. Pidd, M., De Silva, F., & Eglese, R. (1996). A simulation model for emergency evacuation. *European Journal of Operational Research*, 90(3), 413-419.

35. Savelsbergh, M. W., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, 29(1), 17-29.

36. Sayyady, F. (2007). Optimizing the use of Public Transit System in no-Notice Evacuations in Urban Areas. Master Thesis: Mississippi State University.

37. Sayyady, F., & Eksioglu, S. D. (2010). Optimizing the use of public transit system during no-notice evacuation of urban areas. *Computers & Industrial Engineering*, 59(4), 488-495.

38. Sheffi, Y., Mahmassani, H., & Powell, W. B. (1982). A transportation network evacuation model. *Transportation Research Part A*, 16(3), 209-218.

39. Southworth, F. (1991). Regional evacuation modeling: A state-of-the-art review. Oak Ridge National Laboratory, Oak Ridge, TN, USA.

40. Tarantilis, C.D., Kiranoudis, C.T., & Vassiliadis, V. S. (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, 152(1), 148-158.

41. Urbanik, T. (2000). Evacuation time estimates for nuclear power plants. *Journal of Hazardous Materials*, 75(2), 165-180.

42. Urbina, E., & Wolshon, B. (2003). National review of hurricane evacuation plans and policies: A comparison and contrast of state practices. *Transportation Research Part A*, 37(3), 257-275.

43. Xie, C., Lin, D., & Travis Waller, S. (2010). A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies. *Transportation Research Part E*, 46(3), 295-316.

44. Xie, C., & Turnquist, M. A. (2011). Lane-based evacuation network optimization: An integrated lagrangian relaxation and tabu search approach. *Transportation Research Part C*, 19(1), 40-63.

CURRICULUM VITAE

Mojtaba Heydar

Place of Birth: Tehran, Iran

EDUCATION

Ph.D. University of Wisconsin-Milwaukee May 2014
Major: Industrial Engineering—Operations Research
Minor: Transportation Engineering
Dissertation title: Operations Research Modeling of Cyclic Train Timetabling, Cyclic Train Routing, and Bus Routing Problems
Advisor: Matthew E. H. Petering

MSc. Azad University, Tehran South Branch, Tehran, Iran July 2008
Industrial Engineering—Industrial Engineering
Thesis Title: Multi-objective scheduling of flexible manufacturing cells using genetic algorithm

BSc. Azad University, South Tehran Branch, Tehran, Iran July 2005
Major: Industrial Engineering—Production Systems

PUBLICATIONS

Refereed Journal Publications

M Heydar, MEH Petering & D Bergmann, 2013, "Mixed integer programming for minimizing the period of a cyclic railway timetable for a single track with two train types," *Computers & Industrial Engineering*, 66(1), 171-185

S Ebrahimnejad, SM Mousavi, R Tavakkoli-Moghaddam & **M Heydar**, "A new fuzzy compromise approach for ranking risk in mega projects: a comparative analysis," to appear in *Journal of Intelligent and Fuzzy Systems*, 26(2), 949-959

Journal Submissions

MEH Petering, **M Heydar** & D Bergmann, "Mixed integer programming for cyclic train timetabling and routing along a single track line," *Transportation Science*, submitted in 2012

M Heydar, Y Liu, MEH Petering & Jie Yu, "A mixed integer programming and algorithm for strategic evacuation planning with pedestrian guidance and bus routing", submitted in January 2014

Working Papers

M Heydar & MEH Petering, "Cyclic train scheduling and train routing: a mathematical approach,"

Refereed Conference Proceedings

M Heydar, Y Liu & MEH Petering, 2011, "Evacuating highly populated urban zones during emergency: A transit-based solution and optimal operational strategies," *Proceeding of Transport Chicago*, Chicago, USA, 3 June

SS Mousavi, F Ghazi Nezami, **M Heydar** & MB Aryanejad, 2009, "A new hybrid fuzzy group decision making and factor analysis for selecting maintenance strategy," *Proceeding of the 39th International Conference on Computers & Industrial Engineering*, Troyes, France, 6-8 July

M Heydar, SM Moosavi, R Tavakkoli-Moghaddam & SMH Mojtahedi, 2008, "Fuzzy multi criteria decision making method for temporary storage design in industrial plants," *Proceeding of IEEE International Conference on Industrial Engineering and Engineering Management*, Singapore, 8-11 December

Presentations

M Heydar, Y Liu, M.E.H. Petering, A MIP model for strategic evacuation planning with pedestrian guidance and bus routing, INFORMS Annual Meeting, Minneapolis, MN, USA, 6-9 October, 2013

M. Heydar, M.E.H. Petering**,** D. Bergmann, Mixed integer programming for capacity analysis of a single track railway part II: generalized model, INFORMS Annual Meeting, Phoenix, AZ, USA, 14-17 October, 2012

Y Liu, **M Heydar**, MEH Petering, A two-level transit-based evacuation model for planning of emergency in highly populated urban zones, 2nd International Conference on Evacuation Modeling and Management (ICEM 2012), Northwestern University, Evanston, IL USA, 13-15 August, 2012

M Heydar, Y Liu, MEH Petering, An integrated pedestrian guiding and bus routing model for emergency evacuation planning, INFORMS Annual Meeting, Charlotte, NC, USA, 13-16 November, 2011

M Heydar, Y Liu, MEH Petering, Evacuating highly populated urban zones during emergency: A transit-based solution and optimal operational strategies, Transport Chicago, Chicago, USA, 3 June, 2011

TEACHING EXPERIENCE

Primary Instructor, University of Wisconsin – Milwaukee (Spring 2013)

- IND ENG 360 – Engineering Economic Analysis (34 junior and senior students)

Teaching Assistant, University of Wisconsin – Milwaukee (Fall 2012)

- IND ENG 470 – Methods Engineering (lab instructor - 18 senior and graduate students)

HONORS AND AWARDS

- Distinguished Dissertator Fellowship (2013 – 2014 academic year)
  Graduate School, University of Wisconsin – Milwaukee

- Chancellor's Award (2009, 2010, 2011,2012)
  Department of Industrial & Manufacturing Engineering, University of Wisconsin – Milwaukee

- Graduate School Travel Award (Fall 2011, Fall 2012)
  University of Wisconsin – Milwaukee

- Scholarship for rail short course, National Center for Freight & Infrastructure Research & Education (CFIRE), University of Wisconsin – Madison (Fall 2012)

- Young Researchers Group Travel Award (Fall 2008)
  Azad University - South Tehran Branch