STATE OF CALIFORNIA • DEPARTMENT OF TRANSPORTATION

TECHNICAL REPORT DOCUMENTATION PAGE

TR0003 (REV 10/98)

ADA Notice

For individuals with sensory disabilities, this document is available in alternate formats. For information call (916) 654-6410 or TDD (916) 654-3880 or write Records and Forms Management, 1120 N Street, MS-89, Sacramento, CA 95814.

1. REPORT NUMBER	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER

4. TITLE AND SUBTITLE

Integration of Passenger and Freight Rail Scheduling

5. REPORT DATE

02/02/2017

6. PERFORMING ORGANIZATION CODE

7. AUTHOR

Maged M. Dessouky

8. PERFORMING ORGANIZATION REPORT NO.

Project 15-04 / 65A0533 Task Order 010

9. PERFORMING ORGANIZATION NAME AND ADDRESS

University of Southern California
METRANS Transportation Center
Sol Price School of Public Policy
Ralph and Goldy Lewis Hall RGL 216
Los Angeles, CA 90089-0626

10. WORK UNIT NUMBER

11. CONTRACT OR GRANT NUMBER

65A0533

12. SPONSORING AGENCY AND ADDRESS

CA Department of Transporation
CALTRANS Division of Research, Innovation, and System Information
P.O. Box 942873
Sacramento, CA 942873

13. TYPE OF REPORT AND PERIOD COVERED

Final Report
08/15/2015 to 08/14/2016

14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

16. ABSTRACT

In the United States, freight railways are one of the major ways to transport goods from ports to inland destinations. According to an Association of American Railroad’s study, rail companies move more than 40% of the nation’s total freight. Given the fact that the freight railway industry is already running without much excess capacity, better planning and scheduling tools are needed to effectively manage the scarce resources, to cope with the rapidly increasing demand for railway transportation. The integration of freight train and passenger train scheduling and dispatching is an important railroad management task. Leveraging the current railroad trackage to meet the expanding demand in the future is a challenge. The objective of this project is to improve the efficiency of freight trains by reducing their travelling times while maintaining the punctuality of passenger trains. Thus, we propose a decomposition based heuristic that first minimizes the tardiness of passenger trains and second minimizes the travel times of the freight trains. Our proposed algorithm solves the routing and scheduling problem for real world size rail networks efficiently. We perform simulation experiments on an actual rail network in Southern California to test our proposed heuristic approach. The proposed solution reduces the average travel time of freight trains and reduces the average tardiness of passenger trains over other existing approaches.

17. KEY WORDS

18. DISTRIBUTION STATEMENT

This document is available to the public through CALTRANS and METRANS website: https://www.metrans.org/research/integration-passenger-and-freight-rail-scheduling

19. SECURITY CLASSIFICATION *(of this report)*	20. NUMBER OF PAGES	21. COST OF REPORT CHARGED
Unclassified	57	$99,443.47

Reproduction of completed page authorized.

Integration of Passenger and Freight Rail Scheduling

Final Report

METRANS Project 15-04

February 2, 2017

Principal Investigator:

Maged M. Dessouky

Ph.D. Graduate Students:

Liang Liu

Santiago Carvajal

Lunce Fu

Daniel J. Epstein Department of Industrial and Systems Engineering

University of Southern California

Los Angeles, California

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, and California Department of Transportation in the interest of information exchange. The U.S. Government and California Department of Transportation assume no liability for the contents or use thereof. The contents do not necessarily reflect the official views or policies of the State of California or the Department of Transportation. This report does not constitute a standard, specification, or regulation.

Abstract

In the United States, freight railways are one of the major ways to transport goods from ports to inland destinations. According to an Association of American Railroad's study, rail companies move more than 40% of the nation's total freight. Given the fact that the freight railway industry is already running without much excess capacity, better planning and scheduling tools are needed to effectively manage the scarce resources, to cope with the rapidly increasing demand for railway transportation.

The integration of freight train and passenger train scheduling and dispatching is an important railroad management task. Leveraging the current railroad trackage to meet the expanding demand in the future is a challenge. The objective of this project is to improve the efficiency of freight trains by reducing their travelling times while maintaining the punctuality of passenger trains. Thus, we propose a decomposition based heuristic that first minimizes the tardiness of passenger trains and second minimizes the travel times of the freight trains. Our proposed algorithm solves the routing and scheduling problem for real world size rail networks efficiently.

We perform simulation experiments on an actual rail network in Southern California to test our proposed heuristic approach. The proposed solution reduces the average travel time of freight trains and reduces the average tardiness of passenger trains over other existing approaches.

Disclosure

The project entitled “Integration of Passenger and Freight Rail Scheduling” was funded in entirety under this contract to California Department of Transportation. The Principle Investigator of the project was Professor Maged M. Dessouky of the Viterbi School of Engineering at the University of Southern California and was carried out during the period: 08/15/2015 – 08/14/2016. The total amount of funding was $99,975.

Acknowledgement

We would like to thank METRANS for funding this research.

Table of Contents

1. Introduction 9

   1.1 Background 9

   1.2 Problem Description 9

   1.3 Motivation 11

   1.4 Structure of the Report 12

2. Literature Review 13

3. Problem Statement and Formulation 18

   3.1 Network Construction 19

   3.2 Train Movement Process 23

   3.3 Optimization Model 25

4. Solution Procedure 30

   4.1 Passenger Train Scheduling 31

      4.1.1 Passenger Train Priority Assignment 32

         4.1.1.1 Priority Assignment Experiments 34

      4.1.2 Passenger Train Route Construction 37

         4.1.2.1 Route Construction Experiments 41

   4.2 Freight Train Scheduling 42

      4.2.1 Freight Train Scheduling Order 43

      4.2.2 Freight Train Routing Subproblem 44

      4.2.3 Freight Train Priority Assignment Subproblem 46

         4.2.3.1 Freight Train Priority Assignment Experiments 56

5. Experimental Results 59

6. Implementation 64

7. Conclusions and Future Research 65

8. References 66

List of Tables

Table 1. Performance Comparison Between Priority Assignments I-III 35

Table 2. Calibration of Priority Assignment IV on Random Routes 36

Table 3. Calibration of Crossover and Mutation Probabilities 42

Table 4. Objective Comparison Between BFIPC and Opt-Integer with Same Passenger Schedule 58

Table 5. Test Passenger Train Schedule 60

Table 6. Comparison between Optimal-solution and Heuristic-Solution 60

Table 7. Performance Comparison between Complete-Control, Partial-Control, Random-Departure and Uniform-Departure 62

List of Figures

Figure 1. Three Main Distinct Rail Lines in the Los Angeles Area 11

Figure 2. Sample Railway Network Trackage 20

Figure 3. Sample Railway Network 21

Figure 4. Abstract Graph of Sample Network 23

Figure 5. Arrival and Departure Time in Train Movement 24

Figure 6. The Flow Chart of the Solution Procedure 30

Figure 7. Example Route Encoded Gene 38

Figure 8. Example Chromosome of Route Combination 38

Figure 9. Example of the Crossover Operation 40

Figure 10. Example of the Mutation Operation 40

Figure 11. The Flow Chart of the Genetic Algorithm Procedure 41

Figure 12. Example of the Candidate Insertion Position Along a Triple Track Segment 47

Figure 13. Inference Rules in **Step_3** 55

Figure 14. Test Network with Three Stations 59

1. Introduction

1.1 Background

In the United States, the rail industry has been at the forefront of economic growth. The freight rail industry in particular has a long history of being essential to the United States economy. This is not different in today's world with freight trains accounting for about 1,800 million rail tons in 2014, according to the Association of American Railroads. Every year there are more than 100 million tons of goods transferred through the Ports of Los Angeles and Long Beach. Rail transportation is a cost effective way to move cargo from the ports to distant inland destinations. According to the Association of American Railroad's study, rail companies move more than 40 percent of the nation's total freight. As the total quantity of freight increases, the railroad industry expects further increases in demand.

In addition, passenger train travel has also been increasing over the years. In the Los Angeles area Amtrak alone accounted for 1.3 million passengers in 2015. In California, Amtrak accounted for 814.9 million rail passenger miles. All these demands for passenger trains have even attracted the attention of California government officials who passed a bill to invest 42 billion dollars to build more railroad lines in 2008.

Given that the freight industry is already running without much excess capacity, the industry has to either expand or manage its operations more efficiently. Investing in the expansion of the railroad network is an expensive venture. Moreover, in parts like Orange County or Los Angeles, this might be difficult due to space limitation. Thus, better planning and scheduling methodologies become an effective solution to the problems caused by increasing transportation demand under tight capacity constraints.

1.2 Problem Description

Passenger and freight train scheduling are rather different, passenger train schedules are relatively stable and periodic. For example, Amtrak trains from Los Angeles to San Diego leave at several fixed times every day. Therefore, a master schedule of the times for the passenger trains can be developed several months in advance. On the other hand, freight demand is less known and the freight scheduling procedure can sometimes be initiated very close to the time of the departure of the train.

In many regions around the world, it was uncommon for freight and passenger trains to share the same track resources. Therefore, the scheduling for both kinds of trains could be done separately without one impacting the other. However, in the United States, it is very common for both kinds of trains to compete for the same track resources. Furthermore, this is becoming increasingly common globally due to huge increases in both passenger and freight trains. Consequently, a good integrated schedule becomes vital in order to prevent melt-downs of the railroad network. Furthermore, in urban areas like Los Angeles the railroad system is extremely complex with single, double, and even triple track configurations, as well as high traffic zones and different speed limits. To complicate things even more, it is in urban areas that freight trains and passenger trains commonly share the same tracks. However, each type of train has different priorities and characteristics. For example, freight trains usually travel much slower than passenger trains. Furthermore, freight trains on average are much longer in length averaging from 6000 to 8000 feet. This, as expected, can lead to passenger trains experiencing significant tardiness if freight trains are not scheduled effectively. Thus, to minimize train delays and maximize the capacity of the rail network in urban areas, it is important to integrate both passenger and freight scheduling.

This report proposes an integrated scheduling methodology that accounts for both types of trains. We use the rail network from Downtown Los Angeles to Riverside as our test case. Figure 1 shows the three main distinct rail lines in this corridor (the top two lines are served by Union Pacific and the bottom one is served by BNSF). Both Amtrak and Metrolink passenger trains travel through this corridor. Although our integrated scheduling methodology will be general and applicable to a number of rail corridors, we choose the Los Angeles to Riverside corridor as the test case for our methodology, since we currently have a detailed simulation model of the trackage configuration and data on the rail counts for the different types of trains traveling in this area.

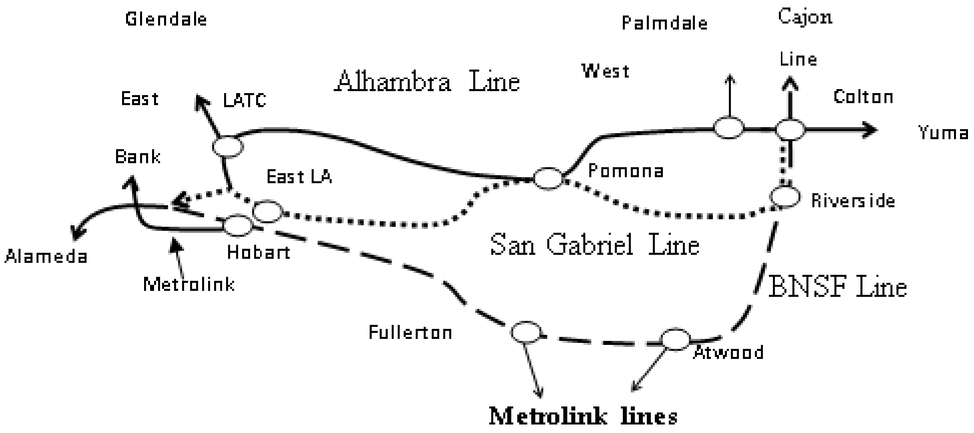


Figure 1. Three Main Distinct Rail Lines in the Los Angeles Area

1.3 Motivation

As mentioned above, the rail industry has always been an essential part of the United States economy. However, in recent years, because of the increase in the demand for both passenger and freight trains, train operations have changed. Especially in Los Angeles, the demand for freight trains has steadily increased due to all the cargo coming into the Ports of Los Angeles and Long Beach. However, Los Angeles is a highly populated city which inhibits the railway network growth. This means that the demand for trains is growing at a much higher rate than what the current railway system can support, and there is a real need to optimize the usage of the railway network so that the demand can still be met so that industries in the area do not need to take their operations elsewhere.

In the past, most freight train scheduling was done by human operators (dispatchers). Each dispatcher was assigned a section of the network that they would monitor. The dispatcher's job was to assign trains a specific route and tell them if they had to decelerate, accelerate or stop, to prevent any collisions or failures in the network. It was then the central dispatcher's job to correct any failures that might occur within sections. As mentioned before this was a viable option because passenger trains and freight trains typically operated at different times and their schedules did not conflict. However, in today's world, because of the increasing usage of the network, more and more freight trains are conflicting with passenger trains. Therefore, just using a manual approach to dispatch trains may not lead to the most efficient schedule. Therefore, we propose an algorithm that can schedule trains efficiently to maximize the usage of the network and minimize passenger train tardiness.

1.4 Structure of the Report

The rest of this report is organized as follows. In Section 2, a literature review of the relevant problems is presented. Section 3 formally defines the network structure and mathematical model. In Section 4, the proposed solution framework is presented. The experimental results of our solution framework are presented in Section 5. In Section 6, we discuss the implementation and applicability of our work. And the conclusions are drawn in Section 7.

2. Literature Review

There have been several survey papers that focus on different aspects of the rail scheduling problem. Cordeau et al. (1998) surveyed optimization models for the train routing and scheduling problem. Caprara et al. (2006) gave a review of strategic, tactical and operational level's decision models for passenger trains. Lusby et al. (2011) reviewed different models and approaches for the train timetabling, train dispatching and train routing problem. This review was grouped into single track network models, general network models and junction routing models. More recently, Harrod (2012) surveyed the train timetabling problem. This survey was organized by problem features (aperiodic or periodic, explicit track infrastructure modeling or not, etc.).

One of the earliest works dates back to Szpigel (1973), where the author related the train scheduling problem on a single track with the job shop scheduling problem. A branch and bound algorithm was proposed to solve this problem. Also experiments were conducted on a single track line with five track sections and ten trains.

Jovanovic and Harker (1991) investigated the railway network with two stations with a lane connecting them and meet points along the lane, where trains can wait, overtake or be overtaken. Arrival and departure times at meet points together with a sequence of trains at meet points were introduced as decision variables to formulate a Mixed Integer Programming (MIP) problem.

Brannlund et al. (1998) formulated a different model for the single track line scheduling problem by discretizing the time periods. Decision variables were indexed by train, track block and time period to indicate that the train occupied the track block at that time. Then a Lagrangian relaxation approach was used to solve the binary integer programming model. Experiments performed on the railway track in the middle of Sweden showed that near optimal schedules were obtained.

Both Jovanovic and Harker (1991) and Brannlund et al. (1998) deal with a single track line railway system. Even though the railway infrastructure is simple, the two models proposed laid the foundation for problems with complex structure and formed two main streams of formulating the train scheduling problem. The main difference between them is that in the model originated from Jovanovic and Harker (1991) times at control points are decision variables. While in the other stream, time is discretized to form a time-space graph. Much research has been done to modify, expand and improve these two basic models.

In the first stream, times at control points are decision variables. Carey and Lockwood (1995) considered the train timetabling problem for an one-way single line railway system, where the decision variables include arrival and departure times at each station and the trains' passing order at each station. They also proposed heuristic strategies to solve the formulated Mixed Integer Programming (MIP) problem. Carey (1994a) extended the model from one-way single line system to a more complex railway system containing multiple railway lines and station platforms. On the other hand Carey (1994b) extended the one-way single track line model to a two-way single line model.

Higgins (1996) proposed a similar MIP formulation to minimize the sum of the weighted delays for each train and the operating cost. Also a branch and bound schema is proposed to solve the MIP, where the lower bound of the optimal solution is estimated through the remaining overtake and crossing delay, in order to reduce the search space in the branch and bound procedure.

Dessouky et al. (2006) also proposed a MIP formulation to minimize the sum of all the trains' arrival times to their destinations. Then a branch and bound scheme, which is based on the idea of fixing the sequence of all the trains passing the same node step by step, is introduced. Numerical experiments are conducted on a complex rail network with comparison between the proposed branch and bound scheme and the CPLEX solver.

Zhou and Zhong (2007) presented a branch and bound procedure to solve the train scheduling problem for a single track line system. In the branching step, they showed that the subproblem can be solved as the longest path problem to determine the earliest start times for each train in different segments. Then for the bounding step, several methods to obtain lower bounds and upper bounds of the optimal solution are proposed. First, a Lagrangian relaxation lower bound is obtained by taking the dual of the relaxation problem. Then another way of obtaining a lower bound is introduced based on the idea of ignoring the existing train conflicts and the potential conflicts generated in the future. To obtain an upper bound, a beam search algorithm is introduced, where only a certain number of search nodes are kept at each level. Numerical experiments show that the lower bounds and upper bounds can reduce the computation time.

Mu and Dessouky (2011) considered the freight train scheduling problem for a complex railway network, where each train may have multiple feasible paths to take. Therefore another binary variable indicating which path to take is introduced to obtain the flexible path formulation. Several heuristic algorithms are proposed to solve the formulated MIP. An intuitive heuristic, which reduces the search space by eliminating “bad” paths, is introduced. Then a genetic algorithm with paths as genes is discussed. Also divide and conquer based heuristics, which either divide up the railway network or group trains into different clusters, are investigated. All the proposed heuristics are compared with a simple look-ahead greedy heuristic and a global neighborhood search heuristic through numerical experiments.

Louwerse and Huisman (2014) considered the problem of adjusting the timetables for passenger trains when railway track disruptions happen. The problem is formulated using an event-activity network. The set of events consists of the arrival and departure events at the stations and of inventory events. The set of activities consists of train activities, headway activities and inventory activities. An Integer Programming (IP) formulation is proposed to decide which trains to cancel and generate a new timetable for the remaining trains with the objective of minimizing the number of cancelled trains and total delay. Numerical experiments are also conducted for the Netherlands Railway system.

Meng and Zhou (2014) investigated the problem of rescheduling and rerouting trains simultaneously on an N-track rail network. A Mixed Integer Programming (MIP) model is formulated to minimize the deviation between the actual arrival time and the planned arrival time. A Lagrangian relaxation schema is proposed to solve the formulated problem, where a label correcting algorithm for the time-dependent shortest path problem is used to solve the routing subproblem in a time-space network. Comparison between the proposed method and a sequential scheduling framework is analyzed through numerical experiments.

In the second stream a time-space graph is constructed and utilized. Capara et al. (2002) considered the train timetabling problem for an one-way single line railway systems. Time is discretized so that a time-space graph is formed where nodes in the graph correspond to departures/arrivals at a certain station at a given time instant. Then an integer linear programming (ILP) problem is formulated to maximize the sum of the profits of the scheduled trains. A Lagrangian relaxation problem is formulated and a subgradient optimization is proposed to obtain near-optimal solutions. A heuristic based on the idea of ranking the trains by decreasing the values of the Lagrangian profit is also proposed to solve the problem.

Capara et al. (2006) extended the previous model by introducing several practical constraints such as manual block signaling between two consecutive stations, maximum number of trains that can be present in a station at the same time, prescribed timetable for a subset of the trains and maintenance operations which keep a track segment occupied for a certain period.

Cacchiani et al. (2010) considered the problem of inserting new freight trains into a railway system where the passenger trains have prescribed timetables. An ILP is formulated with the objective to assign as many new freight trains as possible and also to make their timetables as close as possible to the ideal timetables. A similar Lagrangian relaxation method to Capara et al. (2002) is used and numerical experiments are conducted on large instances to show the efficiency of the proposed method.

Recently, another method of modeling called alternative graph, which can be traced back to Mascis and Pacciarelli (2002), has been developed. In the alternative graph, a set of pairs of directed arcs is specified such that at most one arc from each pair can be selected.The alternative graph is used to solve the train conflict detection and resolution (CDR) problem, where the order of passing one track for two trains is modeled as two alternative arcs in the constructed alternative graph.

Mazzarello and Ottaviani (2007) extended the alternative graph model by considering constraints imposed by the real railway environment. The blocking constraint denotes that a train travelling on a block remains on it until the next block is available. Other constraints in train scheduling are also taken into consideration. The minimum speed constraint enforces that a train must travel at a speed not less than a given speed. The passing constraint imposes a train to pass through a node only after a given time. The minimum dwell time and minimum connection time constraints make sure there is at least a certain time between two train operations. The out of order constraint deals with the scenario where a block is unavailable for a certain time interval. The precedence constraint enforces that one train has higher priority than another train.

D’Ariano et al. (2007) considered the problem of obtaining a new conflict-free timetable when train operations are perturbed. They presented a branch and bound procedure to solve the conflict resolution problem using an alternative graph model, where the upper bound in the branch and bound procedure is obtained by the best value among three heuristics. The first heuristic is to apply a First Come First Serve (FCFS) rule and the second is to apply a First Leave First Serve (FLFS). The third heuristic is a greedy algorithm in which each time chooses the pair of unselected alternatives with the best value and therefore increases the size of the solution iteratively.

Corman et al. (2010) took advantage of the tabu search heuristic to improve the solution quality and computation time for the train conflict detection and resolution problem. Three neighborhood structures for the tabu search are considered. The first is the complete neighborhood containing all the feasible solutions in which only one train follows a different route compared to the incumbent solution. The second is a subset of the first neighborhood with the constraints that trains must have one operation in a backward ramified critical path set (BRCP). The third neighborhood structure extends the second one by considering also the forward ramifications.

3. Problem Statement and Formulation

The objective is to integrate passenger and freight rail scheduling when they share the same trackage to improve the efficiency of freight trains by reducing their travel times while maintaining the punctuality of passenger trains in the same railway network. This objective can be optimized by controlling three kinds of decision variables:

1. Routing decisions: The sequence of track segments that each train travels through.
2. Arrival/Departure time decisions: The time that each train arrives/departs at each track segment.
3. Priority decision: If two or more trains travel on the same track segment, the priority on which train seizes the track segment first.

Considering the structure of a general railway network and the characteristics of the train movement process, the control variables have to be feasible in two aspects:

1. The routes between any two trains should be deadlock free. Deadlock can happen when two or more trains travelling opposite directions request the same resource at the same time.
2. Between any two trains, a minimum safety headway should be guaranteed.

We quantify the freight train efficiency by its travel time, and the passenger train punctuality by its tardiness at its station stops. The freight train travel time is directly related to its delay since delay is typically defined as the difference between the actual travel time when there are other trains in the rail network and its free flow travel time when there are no trains in the network. Passenger train tardiness is defined as the difference between its actual arrival time and scheduled arrival time if the actual arrival is later than the scheduled arrival, else it equals to zero. Intuitively, these two performances are competing with each other since freight trains and passenger trains share the same track resources.

This section first presents the model representation of a generic complex railway network. The abstract network model structure inherits the idea from Dessouky and Leachman (1995) and Lu and Dessouky (2004), which can be used to model a general railway network including single-track lines, double-track lines and triple-track lines. Corresponding to the model structure, the train information consists of origin and destination, speed limit, length and timetable if it is a passenger train. Then we present the mathematical formulation of the model. In order to effectively model and solve this train routing and scheduling problem, we setup metrics to evaluate the freight trains' travel time and the passenger trains' tardiness. We model it as a bi-objective optimization problem, with a set of constraints that guarantee traffic flow conservation, travel time feasibility and a safety headway between the trains.

3.1 Network Construction

To describe the problem using a mathematical model, the actual network should be translated to an arc based network. The actual railway network consists of tracks, sidings, junctions, and platforms. According to their characteristics, we classify the network into two resources: track segments and rail junctions. Track segments are basically segments of track which can be travelled by a train. Rail junctions are used for train crossover movements between track segments. A track segment is a minimum unit in the network, and each segment is represented as a unique resource with one unit of capacity. A rail junction is also represented as a resource with one unit of capacity. By the definition of unit capacity, each segment and each junction can be occupied by at most one train at any time. Here we present the characteristics of the track segments:

a) Each segment has a fixed speed limit. A train follows the minimum of the speed limit of the track segment and the maximum speed of the train. Each junction also has a fixed speed limit (usually smaller) which forces trains to slow down to guarantee a safe switching between segments. Note that the speed limits of consecutive segments are not necessarily the same. Speed limits at stations and curve segments are generally lower.

b) Segments are defined between junctions, which means that there are no junctions that exist within a segment. According to this rule, the simplest way to divide the network is to break it down at all the junctions, and to treat the track between the two junctions as a segment. But this division oversimplifies the network and usually breaks rule (a), because of various speed limits between the two junctions. A more precise division is to break down the network into segments at speed limit changing points. However, a long segment with a constant speed limit would be a waste of the track resource since we assume each segment only has capacity for one train. Therefore, the headway distance between trains would be too large when the segment is too big, which decreases the total capacity of the railway network. Thus, we restrict the length of a segment as no longer than the maximum train length to minimize the unnecessary headway between the trains. A train, however, can occupy several segments simultaneously.

As an illustration of the track segment characteristics, Figure 2 shows a small sample railway network.

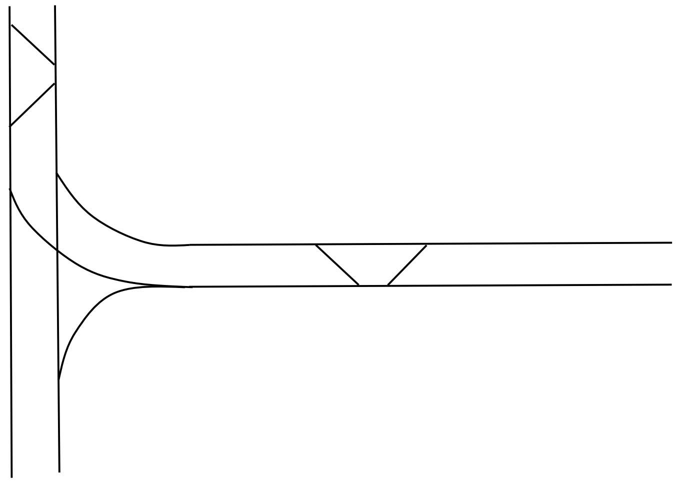
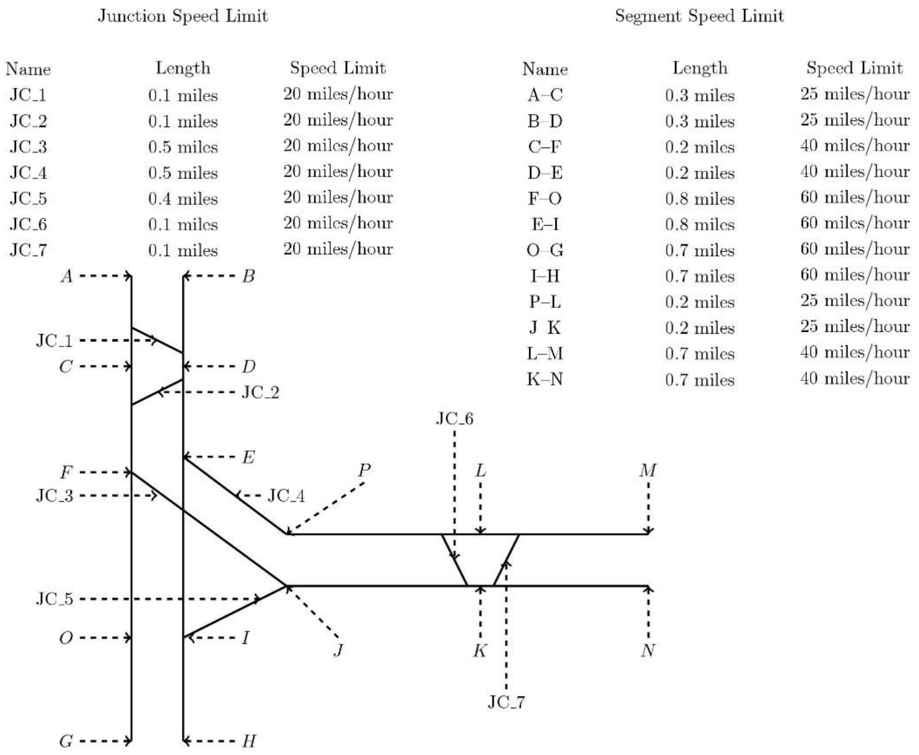


Figure 2. Sample Railway Network Trackage

To translate the sample railway network into our network structure, the rail track is first divided into segments according to the segment characteristics as previously defined. According to our representation, the track segment division is shown in Figure 3. In Figure 3, the network is divided into 12 segments. For example, B-D is a track segment.



Junction Speed Limit

Name	Length	Speed Limit
JC_1	0.1 miles	20 miles/hour
JC_2	0.1 miles	20 miles/hour
JC_3	0.5 miles	20 miles/hour
JC_4	0.5 miles	20 miles/hour
JC_5	0.4 miles	20 miles/hour
JC_6	0.1 miles	20 miles/hour
JC_7	0.1 miles	20 miles/hour

Segment Speed Limit

Name	Length	Speed Limit
A–C	0.3 miles	25 miles/hour
B–D	0.3 miles	25 miles/hour
C–F	0.2 miles	40 miles/hour
D–E	0.2 miles	40 miles/hour
F–O	0.8 miles	60 miles/hour
E–I	0.8 miles	60 miles/hour
O–G	0.7 miles	60 miles/hour
I–H	0.7 miles	60 miles/hour
P–L	0.2 miles	25 miles/hour
J–K	0.2 miles	25 miles/hour
L–M	0.7 miles	40 miles/hour
K–N	0.7 miles	40 miles/hour

Figure 3. Sample Railway Network

The network we use is an arc based graph. The basic components of the network are nodes and arcs.

Node: Each node defines a combination of one or more contiguous segments. The node is the basic component of a complete route, which means that it might contain junctions within it. Note that the railway network itself is an undirected graph, so a track could be entered in any direction. However, given the running direction of a train (e.g., westbound or eastbound), some of the nodes are not enterable. For example, in Figure 3, movements C—F—J—K and D—E—P—L are allowed, but movement P—E—I—H is not allowed. Hence each node is assigned two ports indices, which indicates the entering direction. The two ports are indexed by 0 and 1 respectively. Port 0 is the starting point of travel for a train moving in the node from one direction. Port 1 is the starting point of travel in the opposite direction.

Arc: Arcs are connections between the nodes, and they represent the movement of trains between nodes. Depending on the way we divide the track into segments, the arc may be a junction or not. Arcs are undirected and have zero length. Therefore, the total travel distance of a train equals to the sum of the length of all the nodes that it visits. Sometimes nodes are connected by arcs without junctions. The reason is that if we divide the segment into too few nodes, it generates unnecessary spacing between the trains, since only one train can travel on a node at any time.

Resource: From a modelling point of view, track segments and junctions are represented as a resource in the system, which are utilized by the train entity. A junction itself is a resource. A node may contain several continuous segments, and each is represented as a resource. The distance of a resource on the node is marked as (d_1, d_2), where d_1 is the distance from the end of the segment to port 1, and d_2 is the length of the segment. Each resource has a speed limit, but the speed limit of the junction is not necessarily applied for a train, which depends on whether or not the train utilizes the junction for crossover. In Figure 3, the resource of junction JC_5 is not utilized by the train travelling along E—I—H, since the train does not use the junction for crossover. However, the resource of junction JC_5 is utilized by the train travelling along K—J—I—H. To differentiate these two cases, the resource is marked with a * after the resource name if the junction is needed for the train to pass it, but the speed limit of the junction is not applied since it is not used for crossover.

Based on these definitions, the sample rail network is translated into an abstract graph as shown in Figure 4. There are 11 nodes in the abstract graph, and each node contains one or more track segments and junctions. For example, node N1 contains one segment A—C, node N5 contains two segments F—O and O—G, node N7 contains segment H—I and junction JC_5. A track segment can be included in multiple nodes. For example, track segment H—I is included in nodes N6 and N7. In the abstract graph, the advantage of defining ports on a node is shown. Train movement P→E→I→H is not allowed in reality, and in our abstract graph, the movement of N8→N4→N6 is not allowed since the entering port of N8 is 1 and the entering port of N6 is 0.

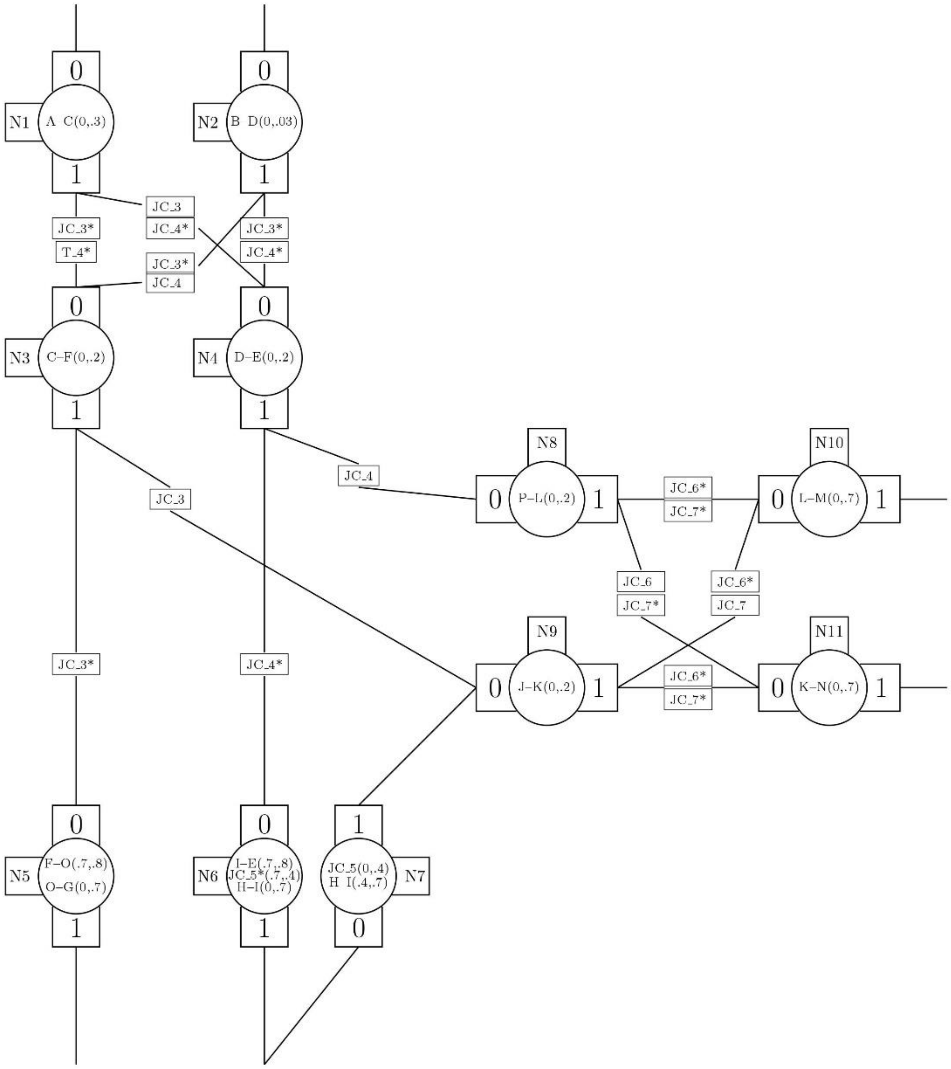


Figure 4. Abstract Graph of Sample Network

3.2 Train Movement Process

The proposed network structure allows for flexible train routing. There are multiple feasible routes for each origin – destination pair. The travel time along the route is influenced by the length and speed limit along the route. The acceleration and deceleration rates of trains may also be considered. In a pre-defined route, the travel time for a train can be expressed as the sum of the travel time of the nodes in the route. We next discuss how to compute the travel time of a train on its route.

During the movement of a train on a node, the length of the train itself is non-negligible because of the division rule of the network. Therefore, the movement of a train cannot be viewed as the movement of a single point. We define three time variables to indicate the position of a train on a node.

a) Arrival time at the beginning of a node: The time when the head of train q arrives to the beginning of node i is denoted as $t_{q,i}^{a}$.
b) Arrival time at the end of a node: The time when the head of train q arrives at the end of node i is denoted as $t_{q,i+1}^{a}$.
c) Departure time at the end of a node: The time when the tail of train q departs from the end of node i is denoted as $t_{q,i}^{d}$.

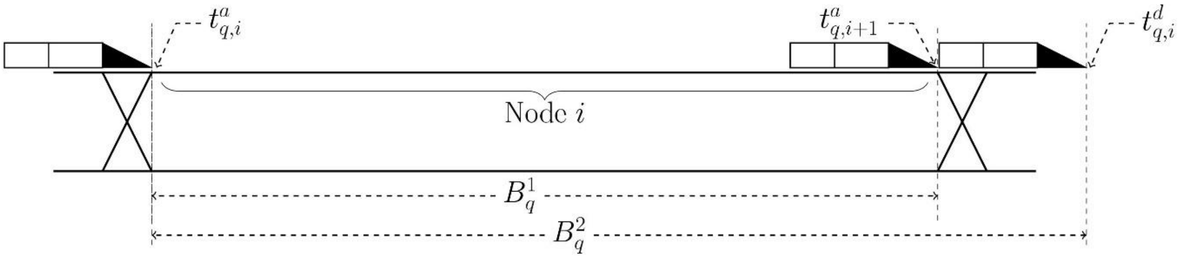


Figure 5. Arrival and Departure Time in Train Movement

Figure 5 gives an example of these time points when a train is travelling through a node. Note that the time points cannot be precisely predicted in advance, because of the congestion of traffic. To evaluate the time that a train spends on a node, we define two measurements.

$$B_{q,i}^{1} = t_{q,i+1}^{a} - t_{q,i}^{a}$$
$$B_{q,i}^{2} = t_{q,i}^{d} - t_{q,i}^{a}$$

The related constraints are that the departure time from a node should not be earlier than the arrival time plus the node travel time. The minimal travel time should be estimated so that the constraints can guarantee the relationship between the arrival and departure times.

The travel time on a node depends on the entering speed v_{i-1} and exiting speed v_i. These two speeds are related to the state of the train on the preceding and succeeding nodes, which are further dependent on the neighboring nodes. In the problem we are studying, trains have flexible routes. Without knowing the complete route, it is difficult to know the entering speed and exiting speed. To get a lower bound on the travel time, we approximate the travel time to make it independent of the entering/exiting speeds. Here we present our estimation methods, which assumes infinite acceleration and deceleration rates.

Under this assumption, the speed of a train can shift to a new speed limit as soon as the speed limit changes. This gives the linear travel time with respect to the length of each segment and train length. Denote the maximum speed of the train as l_0, and length of the train as s_0. Assume that there are n_i^f track segments in node i, each track segment has a speed limit of $l_{k,i}$, and has length of $s_{k,i}, k = 1 \dots n_i^f$. The effective speed limit of segment k on node i is denoted as $L_{k,i} = \min\left(l_{k,i}, l_0\right), k = 1 \dots n_i^f$. The succeeding node of node i is node j, and the speed limit of the first segment on node j is $l_{1,j}$. Note that a node could have multiple succeeding nodes on different routes and $B_{q,i}^2$ depends on $l_{1,j}$, so we modify $B_{q,i}^2$ to be $B_{q,i,j}^2$, and add a dummy end node N_q^{ED} after the last node on the route. The dummy end node has equal length to train q with a speed limit equal to the speed limit of the last segment of the previous node. We then have the following equations.

$$B_{q,i}^1 = \sum_{k=1}^{n_i^f} \frac{s_{k,i}}{L_{k,i}}$$

$$B_{q,i,j}^2 = B_{q,i}^1 + \frac{s_0}{\min(\, l_{1,j}, l_0)}$$

3.3 Optimization Model

We formulate the model as a mixed integer programming problem, which extends the model structure from Dessouky et al. (2006). In their model, only freight trains are considered and the objective is to only minimize the travel time for the freight trains. The objective of our model is to minimize the sum of the freight trains' travel time, and the sum of the passenger trains' tardiness at all the stations. We now formally introduce the notation of the model:

N	Node set of network
Q_f	Set of freight trains
Q_p	Set of passenger trains
$Q = Q_f \cup Q_p$	Set of trains, including freight and passenger trains
O_q	The origin node of train q's schedule
D_q	The destination node of train q's schedule
S_q	The set of station stop nodes along passenger train q's schedule, including origin and destination
N_q^{ED}	The auxiliary dummy end node after destination D_q
$T_{q,s}$	The scheduled arrival time of passenger train q at node s which has a passenger station stop
T^E	The end time of daily operation, which is set to be 23:59
μ	Minimum safety headway between trains
M	A sufficiently large number
$t_{q,i}^a$	The arrival time of train q to node i
$t_{q,i}^d$	The departure time of train q from node i
$I_{q,i,j}$	Binary variable to indicate if train q travels from node i to node j
$x_{q_1,q_2,i}$	Binary variable to indicate if train q_1 passes node i before train q_2

There are three sets of decision variables in the model: $t_{q,i}^a$ and $t_{q,i}^d$ are referred as the *time decisions*. $I_{q,i,j}$ are referred as the *route decisions* and $x_{q_1,q_2,i}$ are referred as the *priority decisions*. The time decisions are related to the time that each train enters or exits the nodes, the route decisions are the set of nodes that each train uses, and the priority decisions are the sequence of trains traveling on each node.

The flexible route assumption in our model enables the trains to travel through multiple routes in the network. For each node along the route, there is a set of decision variables associated with it. The size of the problem grows with the number of nodes. However, given the origin station, destination station and running direction, some of the nodes in the network are not reachable. Before we formulate the complete model, we deploy a route elimination for each train. Note that now the graph is acyclic directed, so we can use a tree searching algorithm such as breadth first search or depth first search. The route elimination step returns the subset of the candidate nodes, which are reachable for each of the trains. For each node in the reduced node set, the succeeding and preceding node sets are also known. The reduced node set for train q is denoted as N_q^t, and the succeeding and preceding node sets are $N_{i,q}^{suc}$ and $N_{i,q}^{pre}$ for $i \in N_q^t$. Similarly, $I_{q,i,j}$ and $x_{q_1,q_2,i}$ are also defined only within the reduced node set of each train. After the route elimination, the size of the problem can be significantly reduced. Corresponding constraints can be defined only on the reduced node set. Our optimization model is presented as follows.

Objective function:

$$Obj1: \qquad \min \sum_{q \in Q_f} (t_{q,D_q}^a - t_{q,O_q}^a)$$

$$Obj2: \quad \min \sum_{q \in Q_p} \sum_{s \in S_q} \max(t_{q,s}^a - T_{q,s}\ ,0)$$

Subject to:

$$\sum_{j \in N_{O_q,q}^{suc}} I_{q,O_q,j} = 1, \forall q \in Q \tag{2}$$

$$\sum_{i \in N_{D_q,q}^{pre}} I_{q,i,D_q} = 1, \forall q \in Q \tag{3}$$

$$\sum_{j \in N_{s,q}^{suc}} I_{q,s,j} = 1, \forall q \in Q_p, s \ \in S_q \backslash \{O_q, D_q\} \tag{4}$$

$$\sum_{i \in N_{j,q}^{pre}} I_{q,i,j} = \sum_{k \in N_{j,q}^{suc}} I_{q,j,k}\,, \forall q \in Q, \forall j \in N_q^t \tag{5}$$

$$\left(1 - I_{q,i,j}\right)M + t_{q,j}^a - t_{q,i}^a \geq B_{q,i}^1, \forall q \in Q, i \in N_q^t, j \in N_{i,q}^{suc} \tag{6}$$

$$\left(1 - I_{q,i,j}\right)M + t_{q,j}^d - t_{q,i}^d \geq B_{q,j}^1, \forall q \in Q, i \in N_q^t, j \in N_{i,q}^{suc} \tag{7}$$

$$t_{q,D_q}^d - t_{q,D_q}^a \geq \ B_{q,D_q,N_q^{ED}}^2\,, \ \ \forall q \in Q \tag{8}$$

$$\left(1 - I_{q,i,j}\right)M + t_{q,i}^{d} - t_{q,j}^{a} \geq B_{q,i,j}^{2} - B_{q,i}^{1}, \forall q \in Q, i \in N_q^t, j \in N_{i,q}^{suc} \tag{9}$$

$$\left(1 - x_{q_1,q_2,i}\right)M + t_{q_2,i}^{a} \geq t_{q_1,i}^{d} + \mu, \forall q_1, q_2 \in Q, i \in N_{q_1}^t \cap N_{q_2}^t \tag{10}$$

$$\left(2 - \sum_{j \in N_{i,q_1}^{suc}} I_{q_1,i,j} - \sum_{k \in N_{i,q_2}^{suc}} I_{q_2,i,k}\right)M + x_{q_1,q_2,i}M + t_{q_1,i}^{a} \geq t_{q_2,i}^{d} + \mu,$$

$$\forall q_1, q_2 \in Q,\ q_1 \neq q_2,\ i \in N_{q_1}^t \cap N_{q_2}^t \tag{11}$$

$$x_{q_1,q_2,i} \leq \sum_{j \in N_{i,q_1}^{suc}} I_{q_1,i,j} + \sum_{k \in N_{i,q_2}^{suc}} I_{q_2,i,k}, \forall q_1, q_2 \in Q, q_1 \neq q_2,\ i \in N_{q_1}^t \cap N_{q_2}^t \tag{12}$$

$$t_{q,O_q}^{a} \geq T_{q,O_q},\ \forall q \in Q_p \tag{13}$$

$$t_{q,D_q}^{a} \leq T^E,\ \forall q \in Q \tag{14}$$

$$t_{q,i}^{a} \geq 0, \forall q \in Q,\ i \in N_q^t \tag{15}$$

$$t_{q,i}^{d} \geq 0, \forall q \in Q,\ i \in N_q^t \tag{16}$$

$$x_{q_1,q_2,i} \in \{0,1\},\ \forall q_1, q_2 \in Q,\ q_1 \neq q_2,\ i \in N_{q_1}^t \cap N_{q_2}^t \tag{17}$$

$$I_{q,i,j} = \{0,1\},\ \forall q \in Q, i \in N_q^t,\ j \in N_{i,q}^{suc} \tag{18}$$

The bio-objective function minimizes both the total travel times for the freight trains (Obj1) and the total tardiness for the passenger trains (Obj2). Constraints (2) - (3) ensure the route of a train has to start from the origin node and end at the destination node. Constraints (4) state that passenger trains have to visit their intermediate station stops. Constraints (5) guarantee the flow conservation on each node. Constraints (6) – (8) ensure the minimum travel time on each node. If the train encounters any waiting such as congestion, the travel time is greater than the minimum travel time which is the free flow travel time. Constraints (9) ensure the minimum travel time for a train to completely clear the occupation of the current node. Constraints (10) – (11) are the deadlock avoidance mechanism that keeps the distance between the trains to be above the minimum safety headway. Constraints (12) force $x_{q_1,q_2,i} = 0$ when both trains q_1 and q_2 do not travel on node i. Constraints (13) state that the departure time of a passenger train from the origin station can not be earlier than the scheduled departure time. Constraints (14) ensure that the train reaches its destination within the daily operation horizon. Constraints (15) – (18) are the domain constraints for the decision variables.

The problem is hard to solve in two aspects: One is that it is a bi-objective problem, and conflicts occur between freight trains and passenger trains when they request the same track resources. A balance has to be made when resources are allocated. In this study, we assign weights to each of the objectives and optimize the weighted objective. The other difficulty comes from the scalability of the problem. The number of integer variables $x_{q_1,q_2,i}$ exist for every pair of trains on every node and the number of integer routing variables explodes as the size of the rail network grows, especially with additional junctions. The exponential growth in the number of integer variables also makes real size problems computationally hard to solve optimally. Thus, we propose a decomposition based solution procedure that vertically decomposes the original problem and then deploys either optimization or heuristic techniques on each of the subproblems.

4. Solution Procedure

For a general railway network, the optimal dispatching time is an NP-hard problem (Lu and Dessouky, 2004). Recall the three decision variables (times, routes and priorities) are computationally hard to solve simultaneously. Thus we develop a heuristic algorithm that decomposes the problem into subproblems to solve problems of realistic size. In Figure 6, the overall logic of the heuristic algorithm is presented.

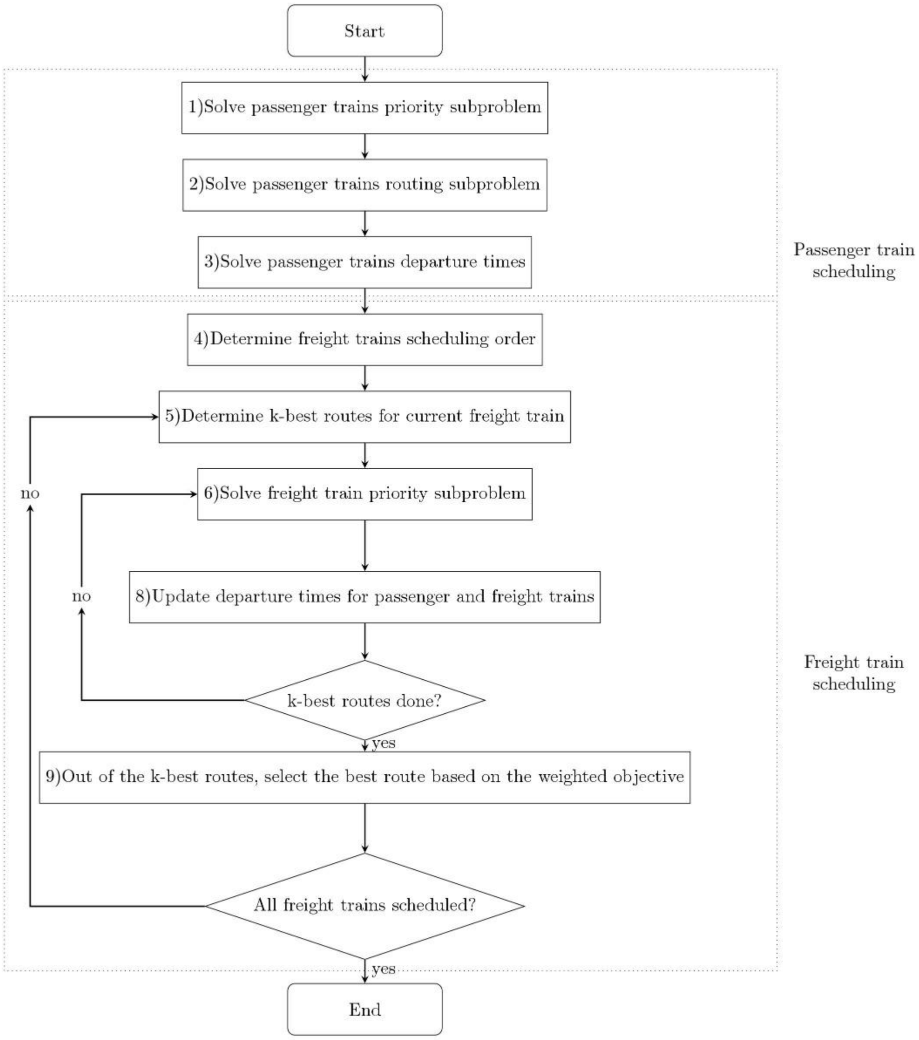

Start
1)Solve passenger trains priority subproblem
2)Solve passenger trains routing subproblem
3)Solve passenger trains departure times
Passenger train scheduling
4)Determine freight trains scheduling order
5)Determine k-best routes for current freight train
no
6)Solve freight train priority subproblem
no
8)Update departure times for passenger and freight trains
Freight train scheduling
k-best routes done?
yes
9)Out of the k-best routes, select the best route based on the weighted objective
All freight trains scheduled?
yes
End

Figure 6. The Flow Chart of the Solution Procedure

The algorithm we propose is based on a two-step decomposition of the original problem. The first decomposition is train based, which decomposes the problem into a subproblem containing only decisions for passenger trains and a subproblem containing only decisions for freight trains. The second decomposition is route based, which decomposes the freight train subproblem into another set of subproblems and solves the priority variable for each route. Specifically, the algorithm is as follows. First, passenger trains are scheduled together by solving three subproblems, which are passenger train priority assignment subproblem in Step 1, passenger train routing subproblem in Step 2 and passenger train departure time subproblem in Step 3. The heuristic algorithms for these subproblems are presented in Section 4.1. Second, freight trains are scheduled sequentially according to a predefined scheduling order, and the freight train scheduling order is defined in Step 4. For each of the freight trains, the k-best routes are identified based on the current traffic condition, which is implemented in Step 5. Each of the routes are then evaluated with the best priority assignment in Step 6. After the priority assignment of the freight train, the departure times of all the passenger and freight trains are updated by solving a linear programming problem in Step 7. The details of the freight train scheduling subproblem are presented in Section 4.2.

In summary, our algorithm employs a decomposition based hybrid heuristic. First, the train schedules are vertically decomposed into two phases, passenger train scheduling and freight train scheduling. In the passenger train scheduling phase, only the objective related to passenger tardiness is considered. In the freight train scheduling phase, the weighted objective is considered and we solve the scheduling of freight trains in an iterative approach. We update the passenger train schedules to reduce the freight train travel time if the weighted objective can be improved.

4.1 Passenger Train Scheduling

The objective of this step is to construct a schedule such that there is no tardiness for any passenger train at any station if possible. We maximize the earliness of passenger trains in order to maximize the slack when freight trains are scheduled later. The earliness of a passenger train is defined as the difference between the actual arrival time and the scheduled arrival time. Note that the earliness is negative when a train is tardy.

First, we construct a Max-Min Earliness subproblem related to passenger trains. The objective of this problem is to construct the routes and priorities of passenger trains such that the minimum earliness of the passenger trains in each of the stations is maximized, thus making passenger trains arrive as early as possible. Here maximizing the minimum earliness is preferred rather than maximizing the sum of the earliness, because the former gives a relatively uniform earliness, thus avoiding an unbalanced schedule in terms of on-time performance. The mathematical formulation is expressed as follows.

Subproblem-1:

Objective function:

$$\max \min_{q \in Q_p, s \in S_q} T_{q,s} - t_{q,s}^{a}$$

Subject to:

Constraints (2)-(18) are from the original problem in which only the passenger train set Q_p is considered.

4.1.1 Passenger Train Priority Assignment

To solve the Max-Min Earliness **Subproblem-1**, we first identify some heuristic rules to solve for the priority decision variables. Since the routes of the passenger trains are flexible, the heuristics rules for priority assignment should be general such that they can fit any of the routes for each passenger train. We propose four priority assignment heuristic rules and compare them through experiments.

Priority Assignment I: An intuitive assignment rule to make for the priority decision is to follow the timetable order, which means that the train with an earlier scheduled departure time should have a higher priority compared to another train with a later scheduled departure time at all the nodes along the routes. This method works fine when the schedules are less dense. However, in general, there are drawbacks when we directly deploy this method. The actual departure time is unknown. The timetable only shows the scheduled departure time, but the actual departure time is not guaranteed to be on-time because of congestion. When a late departure occurs but priority is still assigned based on the timetable, it tends to make some trains become very tardy.

Priority Assignment II: Another assignment rule is to estimate the arrival time of the train at the nodes along its route, and allocate the track resources to trains on a first-come-first-serve (FCFS) order. Train q's earliest possible arrival time at some node i in its routes is calculated as follows. The complete candidate routes set for train q is R_q. We denote $t^a_{q,i,r}$ as the arrival time of train q at node i through route r. In route r, the node set before node i is denoted as $N^c_{q,i,r}$.

Objective function:

$$\min T^a_{q,i}$$

Subject to:

$$t^a_{q,O_q,r} \geq \ T_{q,O_q}, \forall r \in R_q \tag{19}$$

$$t^a_{q,j+1,r} \geq t^a_{q,j,r} + B^1_{q,j}, \ \ \forall \ j \in N^c_{q,i,r}, \ \ r \in R_q \tag{20}$$

$$T^a_{q,i} \geq t^a_{q,i,r}, \ \ \forall r \in R_q \tag{21}$$

The priority at node i is then based on the ranked order of $T^a_{q,i}$ from smallest to largest for all q. Since the actual departure time and traffic congestion is unknown, the arrival time at any given node cannot be precisely estimated. The solution to the problem gives us the earliest arrival time $T^a_{q,i}$ under an arbitrary route, with the assumption of an on-time departure and zero congestion (free flow travel time). If the candidate route sets for trains are disjoint in time and space, this assignment rule works fine because the FCFS rule allocates the track resources to trains as early as possible. This would lead to passenger train running times to be greatly reduced; however, this assignment rule does not consider the anticipated tardiness of passenger trains at its next station. Intuitively, some of the tardiness can be avoided if a higher priority is assigned along the route of a tardy passenger train.

Priority Assignment III: In this assignment rule, we assign the priority according to the minimal anticipated tardiness at the next station. The minimal anticipated tardiness can be solved through a similar model as in Priority Assignment II, with only a slight transformation. We denote the minimal anticipated tardiness of train q at node i as $T^t_{q,i}$, where $T^t_{q,i} = T^a_{q,i} + T^b_{q,i,s} - T_{q,s}$. $T^b_{q,i,s}$ is the shortest free flow travel time of train q from node i to node s that contains the next station and

$T_{q,s}$ is the scheduled arrival time at s. Note that if the train is early, the anticipated tardiness value is defined to be negative. The priority is then based on the ranked order of $T^t_{q,i}$ from largest to smallest for all q at node i. The advantage of this assignment rule is that it directly takes the anticipated tardiness into account. The disadvantage is that the tardiness measure is very sensitive to the time estimation. If the time estimation is not accurate, trains may unnecessarily be held back.

Priority Assignment IV: In most of the cases, the FCFS rule in Priority Assignment II minimizes the waiting times of the trains. However, when some of schedules are tighter than the others, Priority Assignment III considers the tardiness and increases the priority of trains that are running late. A hybrid assignment rule is to combine Priority Assignment II and Priority Assignment III. This priority assignment is based on the weighted average of the arrival time and the anticipated tardiness. We assign priority mainly according to a FCFS rule, but when the anticipated tardiness is detected to be too large, the train's priority increases. This method considers the anticipated tardiness, but does not necessarily hold a train back. We define a metric to measure the weighted sum of the earliest arrival time $T^a_{q,i}$ and anticipated tardiness $T^t_{q,i}$, as a weighted sum time $T^w_{q,i}$:

$$T^w_{q,i} = \alpha * T^a_{q,i} - (1 - \alpha) * T^t_{q,i}, \ \ \forall q \in Q_p \tag{22}$$

A smaller weighted sum $T^w_{q,i}$ gives the train a higher priority to pass node i. The weighted coefficients are tuned in the following experiments.

4.1.1.1 Priority Assignment Experiments

The performance of the priority assignment rules depends on the network structure and the density of the passenger train schedules. Experiments should be done to evaluate each assignment rule on a specific network and schedule. We use part of the railway network for the Los Angeles area, from Downtown to Fullerton. In these experiments, we only consider the daily passenger trains. The test network contains about 20 miles of track, including double-track and triple-track segments. Specifically, there are 77 track segments and 31 junctions in total. In our abstract graph, there are 69 nodes and 62 arcs. The daily passenger train schedule contains 51 passenger trains in total in both directions. There are a total of four passenger train station stops in this area. This same network is also used in the rest of this report to test the other aspects of the solution procedure.

Recall that the earliest arrival time $T_{q,i}^{a}$ and minimal anticipated tardiness $T_{q,i}^{t}$ are defined under the complete candidate route set R_q, which contains all the routes between the origin station O_q and the current node i. However, the number of routes grows exponentially with the number of junctions. In this sample network and schedule, **Subproblem-1** has 138,085 integer variables, hence it is computationally prohibitive to solve the problem to compute $T_{q,i}^{a}$ and $T_{q,i}^{t}$ for every q, i combination considering all possible routes. Thus we randomly select a route for each passenger train q and we compute $T_{q,i}^{a}$ and $T_{q,i}^{t}$ for every node i along the selected route. To compare the performance of the Priority Assignment rules I-IV, we perform 1,000 random samples for the passenger trains route combination, and evaluate the objective of **Subproblem-1**, which is computed by fixing the passenger trains' routes as the sampled routes, setting the priorities based on the priority assignment rule, and solving the arrival/departure times as a linear programming problem. Since by fixing the routing variables and priority variables, **Subproblem-1** reduces to a linear programming problem. Table 1 records the average Max-Min Earliness of the 1,000 random samples. Note that a negative Max-Min earliness value indicates that all the passenger trains are tardy. Table 2 presents the sensitivity analysis on the weighted coefficient in Priority Assignment IV.

Priority Assignment Rule	Average Max-Min Earliness
I	-30.453
II	-16.616
III	-120.834

Table 1. Performance Comparison Between Priority Assignments I-III

Coefficient	Max-min Earliness	
α	Average	Max
0.99	-14.370	1.098
0.95	-15.766	3.426
0.92	-13.668	1.924
0.9	-13.065	3.139
0.87	-15.257	2.975
0.85	-13.035	1.825
0.83	-14.264	0.379
0.8	-16.211	0.010
0.7	-24.042	-1.415
0.6	-21.752	-0.470
0.5	-31.306	-13.007
0.4	-39.397	-31.259
0.3	-65.031	-48.988

Table 2. Calibration of Priority Assignment IV on Random Routes

From the results, we can conclude that for the random routes, Priority Assignment IV outperforms the other three priority assignment rules for our test network and timetable. For this reason, we employ Priority Assignment IV with a coefficient setting of $\alpha = 0.9$ to decide the relative priority of passenger trains in the following section.

The primary reason that all priority assignment rules have a negative average Max-Min Earliness is because the routes are randomly generated. The randomly selected route combination does not provide any guarantee to a low traffic congestion or a selection of a route with the least free flow travel time, thus making the average Max-Min earliness to be negative. However, the Max column in Table 2 indicates that at least some of the route combinations can generate positive Max-Min earliness, which means zero tardiness for all the passenger trains. The next step in solving the Max-Min Earliness subproblem is to construct a route combination between passenger trains such that the Max-Min Earliness is positive and maximized.

4.1.2 Passenger Train Route Construction

For a general railway network, the routing of trains is flexible. Shorter distances and higher speed limits are preferred when a route is selected. Due to the fact that the track resources are shared by multiple trains, assigning similar routes to different trains may generate extra congestion. Therefore, route construction should balance individual train travel time and network congestion. In this section, we present a genetic algorithm based heuristic to construct the routes for passenger trains.

The genetic algorithm is a search heuristic that mimics the process of natural selection. It is widely used to solve many optimization problems. In a genetic algorithm, an evolution process is performed on a population of candidate solutions to guide the search towards a better solution. A typical genetic algorithm requires a chromosome representation of candidate solutions and a fitness function to evaluate the solutions. First, we define the chromosome representation and then we introduce the fitness function.

Population initialization: A chromosome is defined as a bit string that represents the route combinations of all passenger trains. Each bit of the chromosome is mapped to a route of a specific train. The routes domain of a train is defined under some rules so that the space of the chromosome is limited to a reasonable size. For each passenger train, the total number of routes that are generated is restricted to a maximum of n^c, so that each bit of the chromosome is an integer with a range of $[1, n^c]$. There are $|Q_p|$ passenger trains so that the length of the chromosome is $|Q_p|$. In Figures 7 and 8, an example of chromosome-routes mapping on a segment of a double-track rail network is shown. In the example, we set $n^c = 2$ and $Q_p = \{q_1, q_2, q_3\}$

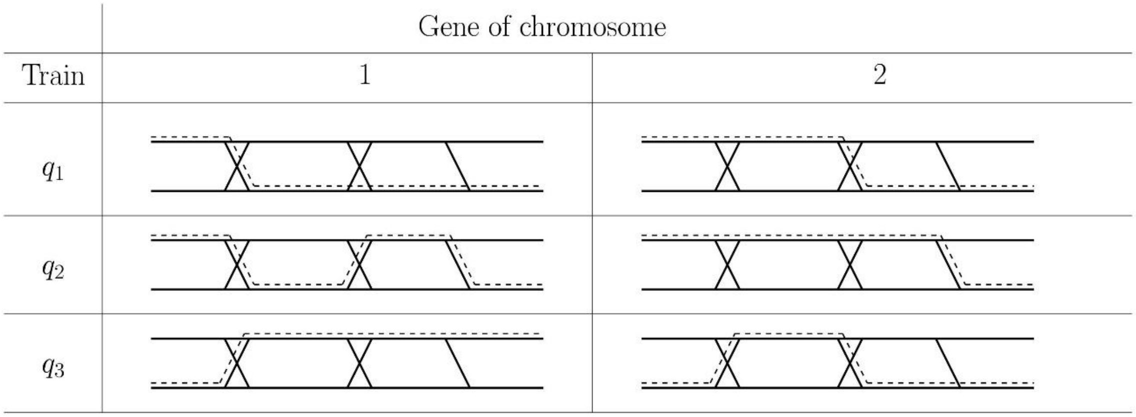


Figure 7. Example Route Encoded Gene

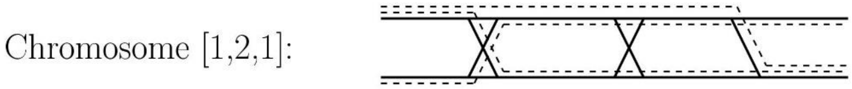


Figure 8. Example Chromosome of Route Combination

Note that the number of possible routes between the two stations grows exponentially with the number of junctions along the track. Therefore, the candidate route set is limited to the size of n^c for each train. The criteria to construct the candidate route set determines the overlap of the routes between the trains and influences the congestion along the track. Here we introduce three criteria in the candidate route set construction:

1. Trains that travel in the same direction prefer to use the same track. Two trains that travel on exactly the same route but in the opposite directions generate extreme tardiness for the later train, since it has to wait until its beginning node is released by the other train. Given that many of the tracks are either double track or triple track segments, the first criteria is that trains prefer to use the left most track if possible.
2. Trains are not allowed to make frequent track changes during their travel. The speed limits of junctions are usually much lower than that of normal track, for safety reasons during the crossover. Frequently crossing the junctions adds additional travel time. Therefore, in our algorithm, each passenger train is only allowed to do one track change every 5 times its length.
3. After changing track to the non-left track, the train changes back to its preferred track (left track) within 3 possible junctions. Without a recovery mechanism, the train could possibly occupy the preferred track of another train travelling in the opposite direction.

The candidate route set is filled randomly according to the criteria above until n^c distinct routes are generated.

Fitness function: The fitness value is defined as the objective value of the Max-Min Earliness subproblem. The route combination is retrieved from a given chromosome and then routes and priority integer variables are substituted in the Max-Min Earliness subproblem. Since after the priority decision and route construction, all integer variables are fixed, the problem is now a linear program and can be solved efficiently. Note that constraints (14) state that all operations should be completed within a daily period. Thus it is possible that some of the chromosomes yield infeasible solutions. For example, if the scheduled arrival time of a passenger train is near the end of the day, a large tardiness makes **Subproblem-1** infeasible. In this case, the objective value is set as $-\infty$ so the chromosome is eliminated in the selection process. The fitness function solves the Max-Min Earliness subproblem and obtains the arrival and departure times of each passenger train at each node. The arrival and departure times for each train at each node are recorded, and they are used in the scheduling of freight trains in the next section.

The basic operations of the genetic algorithm consists of *selection*, *crossover* and *mutation*, which are defined as follows. This generation process is repeated until a termination criterion is met.

Selection: Three selection methods are implemented: top-K, roulette wheel and tournament selection. In the top-K selection method, we sort the new population according to the fitness values, and return the top K individuals as the new generation. The Roulette method first sorts the population, then randomly appends the m^{th} highest individual to the new generation. The tournament randomly picks several 3-individual groups, and adds the individual with the highest fitness value from a group to the new generation. Preliminary experiments show that the tournament method performs best to find the individual with the highest fitness value.

Crossover: The crossover step generates new route combinations from the parent generation with probability θ_c. A two-point crossover is then executed to swap the bit segments between the two parents. Two randomly continuous bit strings of the same size from the parents are extracted and exchanged, and consequently two children are generated. Figure 9 shows an example of a crossover. The bit string that is selected is from the 3^{rd} to the 6^{th} string.

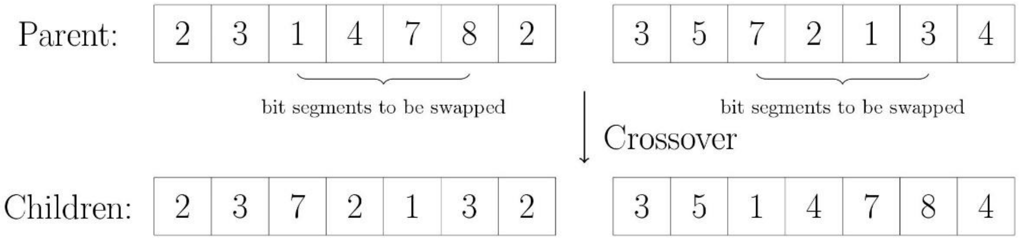


Figure 9. Example of the Crossover Operation

Mutation: The mutation operation is performed after the crossover step with probability θ_m. It imitates the mutation as seen in natural evolution, which adds the variation to chromosomes in generating the children. Mutation happens randomly with a small probability and changes the chromosome into a feasible neighborhood solution. We define the neighborhood solution as being the current solution, with the exception of one route. Each time when a mutation is triggered, a train changes its route to another one within the candidate route set. The mutation uses a single point mutation strategy so that each mutation changes a random bit in the chromosome. Figure 10 presents an example of the mutation operation.

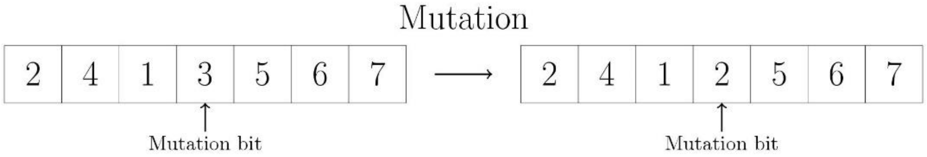


Figure 10. Example of the Mutation Operation

Termination criteria: The termination criteria is met when either one of the following two conditions is satisfied.

a) The maximum minimum earliness e^M is reached.
b) The maximum number of generations n^G is reached

Figure 11 summarizes the flow chart of the genetic algorithm. Note that in the fitness valuation step, the route combination from the chromosome and the priority decision from Priority IV Assignment Rule are fed to **Subproblem-1**. We then solve for the arrival and departure time decisions and the objective value of **Subproblem-1**.

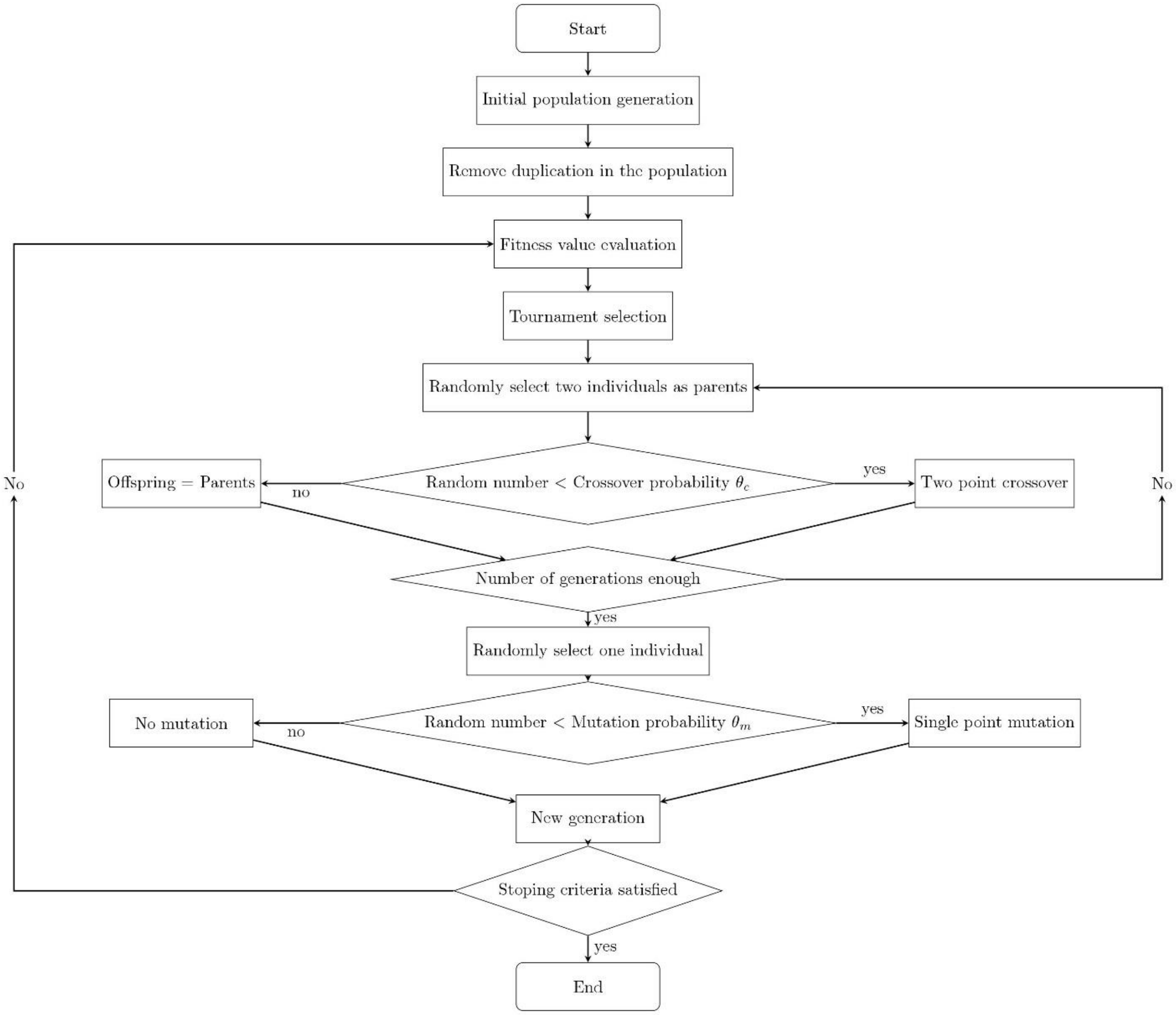

Start
Initial population generation
Remove duplication in the population
Fitness value evaluation
Tournament selection
Randomly select two individuals as parents
Offspring = Parents
no
Random number < Crossover probability θ_c
yes
Two point crossover
No
No
Number of generations enough
yes
Randomly select one individual
No mutation
no
Random number < Mutation probability θ_m
yes
Single point mutation
New generation
Stoping criteria satisfied
yes
End

Figure 11. The Flow Chart of the Genetic Algorithm Procedure

4.1.2.1 Route Construction Experiments

We use the same sample network as in Section 4.1.1.1 to perform experiments in selecting the best combination of crossover probability θ_c and mutation probability θ_m. The total population size is set to be 200, n^c is set to be 20 and the maximum number of generations n^G is set as 100 and $e^M = 16.77$. We compute this e^M value by first identifying for each train its maximum earliness along all its stops and routes with the assumption of on-time departure and zero congestion on the route (free flow travel time), and then e^M is set to be the minimum of these values across all the trains. Table 3 shows the Max-Min Earliness for different combinations of θ_c

and θ_m. For each combination, we run it five times and record the best value. The results in Table 3 use the genetic algorithm for routing and Priority Assignment IV rule with $\alpha = 0.9$. According to this result, $\theta_c = 0.5$ and $\theta_m = 0.2$ are the best settings for the parameters. The result of 5.41 for the Max-Min Earliness compares favorably with the average and maximum results in Table 2. Clearly the genetic algorithm provides solutions that outperform the average of the 1,000 random sampled routes, but it also outperforms the maximum value of these random samples.

Max-Min Earliness				
θ_c θ_m	0.1	0.3	0.5	0.7
0.1	-0.41	1.17	1.17	-0.41
0.2	2.31	3.67	5.41	2.31
0.3	1.87	1.17	1.87	1.87
0.4	1.17	-0.41	-1.11	-1.11

Table 3. Calibration of Crossover and Mutation Probabilities

4.2 Freight Train Scheduling

Unlike passenger trains, freight trains have more flexibility in selecting their departure times from their origin stations, as long as the arrival time to the destination is on time. The routing of freight trains is also flexible, but it should strive towards having minimum impact on the rail traffic along the corridor. Freight train scheduling cannot follow the same strategy used for passenger trains because of the following reasons:

1. The priority assignment rules are different. Priority assignment rules for passenger trains are determined by the anticipated arrival time and tardiness.
2. The route requirements are different. The route construction for freight trains should be based on adding minimal traffic congestion to the corridor/network. The routes construction method developed for passenger trains focused only on maximizing the minimum earliness and not on the travel time minimization. Thus, a different routing procedure that minimizes the travel time and takes into account network congestion effects needs to be applied for freight trains.

3. The priority relationship between freight trains and passenger trains should be balanced. When freight trains are scheduled, the freight trains' travel time and the passenger trains' tardiness should be balanced when the priority relationship is assigned.

In this section, we present our freight train scheduling algorithm. Freight trains are sequentially scheduled into the system. In this algorithm, the freight trains' travel time and passenger trains' tardiness are balanced each time a new freight train is scheduled. The algorithm contains three steps, which consists of three subproblems: freight train scheduling order, freight train routing and freight train priority setting.

4.2.1 Freight Train Scheduling Order

In the algorithm, freight trains are scheduled sequentially. Once a freight train is scheduled, its route and priority are fixed. First, the freight trains are grouped according to their origin-destination pair. Each group of trains is defined as a freight train *demand set*. Note that the trains in a demand set share the same origin and destination, thus the similarity of the routes between them is high. Scheduling all the freight trains within a demand set at the same time adds significant amount of traffic on portions of the network, which may cause congestion. The rule is based on evenly spacing the release of the freight trains from the same demand set.

Given K freight train demand sets $\{D_1, D_2 \dots D_K\}$ and each demand set D_k has $|D_k|$ freight trains. First, the K demand sets are sorted in descending order according to their size. Thus, trains in demand sets that are larger (larger $|D_k|$) are more likely to be released earlier than freight trains in smaller demand sets. Then in each demand set D_k, a fixed portion of trains with size of $\lceil \gamma * |D_k| \rceil$ are selected to be scheduled. After a train is scheduled, it is removed from its demand set. The process iterates through the K demand sets until all the sets are empty. For example, if the sorted demand sets are $D_1 = [q_1, q_2, q_3, q_4]$, $D_2 = [q_5, q_6]$, $D_3 = [q_7]$ and $\gamma = 0.5$, then the scheduling order is $[q_1, q_2, q_5, q_7, q_3, q_6, q_4]$.

We use an iterative approach to schedule freight trains. In this approach, first the route of the freight train is assigned based on the current traffic congestion, then the priority relation of the train with all the previously scheduled trains are identified. Finally its route and priority decisions are fed into a linear program to solve for the optimal arrival/departure times. The scheduling order of freight trains does not guarantee the global optimality since it is constructed heuristically. However, it is an efficient method and we show that this scheduling order can give a high quality solution in the experiments.

4.2.2 Freight Train Routing Subproblem

In order for a freight train to be scheduled, its route is first identified. The principle for identifying the freight trains route is based on the balance of its travel time and overall traffic congestion. Note that the traffic congestion of a network depends on the time that the train enters the rail network.

The freight trains can depart from their origins and enter the network at any time throughout the day. However, the optimal route could be different at different times of the day. For example, during the rush hour of the day, the optimal route is highly influenced by the traffic congestion. In our solution approach for freight train scheduling, the routing and priority decisions are made first and then the departure times are identified. Thus, the routing decision for the newly scheduled freight train q^* should rely on an approximation of the traffic congestion. Note that in the train movement, its actual travel time will be greater than the free flow travel time if there is congestion. The earliest arrival time is the departure time at the origin station plus the free flow running time. The delay time, which is the difference between actual arrival time and earliest arrival time, can be used as an indicator for traffic congestion. The delay time at a node is the sum of the delay times of the previously scheduled trains, which can be obtained from the departure/arrival times recorded in the previous iterations.

The previously scheduled trains are denoted as $Q^{\Phi} = Q_f^{\Phi} \cup Q_p,$ in which Q_f^{Φ} represents the previously scheduled freight trains. For a previously scheduled train, let the sequence of nodes along its route be $Z_q, \forall q \in Q^{\Phi}$ and its arrival time at node i is recorded as $t_{q,i}^a, i \in Z_q$. For each node $i \in Z_q$, the corresponding train's congestion factor $c_{q,i}$ can be represented by the difference between the actual arrival time and the earliest arrival time. The earliest arrival time assumes free flow travel (without traffic congestion) from the start node of movement. We denote the free flow travel time of train q from node n_0 to node i through route Z_q as $t_{q,n_0,i}^f$, in which n_0 is the start node of movement for train $q \in Q^{\Phi}$. For the freight train, we consider n_0 to be its origin station.

For the passenger train, we consider n_0 to be the previous node which contains a station stop before node i. The *congestion factor* is calculated as follows:

$$c_{q,i} = \max\left[t^{a}_{q,i} - \left(t^{f}_{q,n_0,i} + t^{a}_{q,n_0}\right), 0\right], \forall q \in Q^{\Phi}, i \in Z_q \tag{23}$$

Additionally, the tardiness of the passenger trains is also included as a term with a weight of ω, meaning that the candidate route selection receives a penalty if it adds to the passenger trains tardiness. The *tardiness factor* of passenger train q at node i is denoted as $h_{q,i}$, and it is defined as the difference between its actual arrival time at node s that contains the next station stop from node i and the scheduled arrival time at s if it is tardy, else it equals to zero.

$$h_{q,i} = \max(t^{a}_{q,s} - T_{q,s}, 0), \forall q \in Q_p, i \in Z_q \tag{24}$$

The total congestion on node i is the sum of the congestion factors of all trains in Q^{Φ} on node i. The total tardiness on node i is the sum of the tardiness factors of all trains in Q_p on node i. We define the total weight of node i as F_i as follows.

$$F_i = \sum_{q \in Q^{\Phi}} c_{q,i} + \omega * \sum_{q \in Q_p} h_{q,i}, \forall i \in N \tag{25}$$

Preliminary experiments show that $\omega = 10$ gives a balance between the total congestion and the total tardiness. The weighted objective value of node i, F_i, is a nonnegative number for evaluating the traffic congestion. A large value means that the potential traffic on node i tends to be heavy. The route selection for the freight train prefers the least congested route, which keeps the travel time short and also adds the least delay/tardiness to the previously scheduled trains. Thus, the best route is defined to be the one with the least travel time under the anticipated traffic. However, the best route may not be the optimal route since the optimality depends on the train scheduling order. To increase the search space, we select the k-best routes with least anticipated traffic by solving the following problem.

For the new scheduled freight train $q^* \in Q_f \backslash Q^{\Phi}_f$, the search space for its route is within the sub-network that only contains the reduced node set N^t_q. We build another network with the same structure of the sub-network, but the weight of each node is assigned as $F_i + B^1_{q^*,i}$, in which $B^1_{q^*,i}$

is the free flow travel time of train q^* on node i, and F_i is the weighted objective which approximates the anticipated traffic at node i. Finding the best k routes for this network is actually finding k shortest paths. This problem can be solved efficiently by deploying the generalized Dijkstra Algorithm (1959) or Eppstein's Algorithm (1998).

4.2.3 Freight Train Priority Assignment Subproblem

The route defines the sequence of nodes that a train travels through. On each node along the route, the priority relationship of the newly scheduled train q^* with the previously scheduled trains should be selected carefully. In this section, we present an algorithm to assign the priorities to the nodes along a given route for a newly scheduled freight train. In Section 4.2.2, the k-best routes are selected to be evaluated and a priority assignment is determined for each of these routes. The selected route from these k-best routes for q^* is the one that minimizes a weighted objective function of the total freight train travel time and the total passenger train tardiness.

We define the insertion position on node i for train q^* as the position in the priority sequence of the previously scheduled trains on node i. Figure 12 shows an example of candidate insertion positions on a node. Assume for node i in the sample network, the priority relationship between the previously scheduled trains are $q_1 > q_2 > q_3$, which means that the given priority decisions are $x_{q_1,q_2,i} = x_{q_1,q_3,i} = x_{q_2,q_3,i} = 1$. Then there are four candidate positions for the new train. For example, position 1 is the newly scheduled train q^* travels through node i before q_1, and position 2 means train q^* travels through node i after q_1 and before q_2. The objective of the freight train priority assignment is to find the insertion positions on the nodes along the route, with a minimum increase in total freight train travel time and total passenger train tardiness.

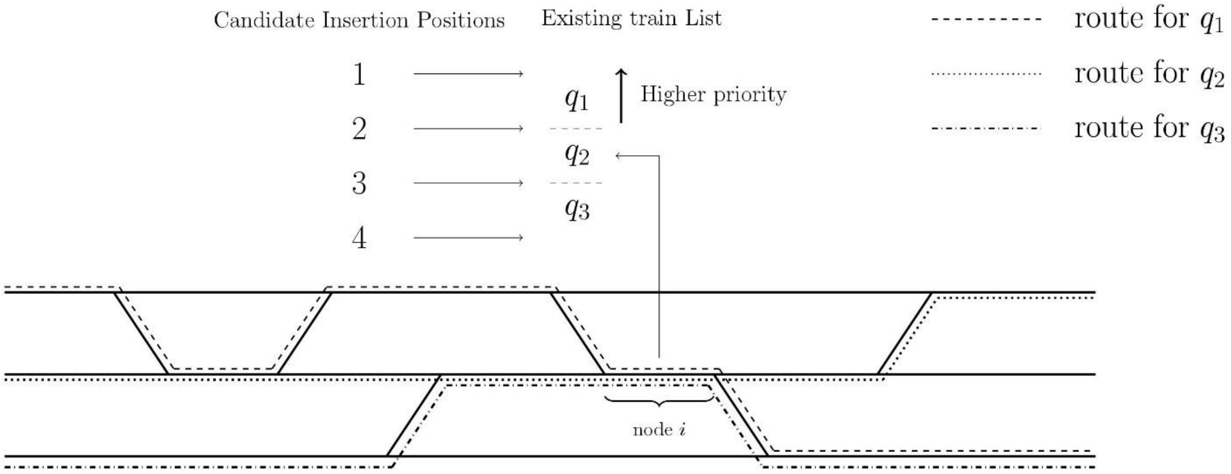


Figure 12. Example of the Candidate Insertion Position Along a Triple Track Segment

We denote the newly scheduled train as q^* and the sequence of nodes along its route as $Z^r_{q^*}$, where route r is one of the k-best routes from Section 4.2.2. We identify an insertion position for train q^* for each node in each route r. For simplification we drop the superscript from $Z^r_{q^*}$ because the insertion procedures are the same for each of the k-best routes. To select the best insertion positions, we define **Subproblem-2**.

Objective function:

$$\min \beta \sum_{q \in Q_f} (t^a_{q,D_q} - t^a_{q,O_q}) + (1-\beta) \sum_{q \in Q_p} \sum_{s \in S_q} \max(t^a_{q,s} - T_{q,s}, 0)$$

Subject to:

Constraints (2)-(18) are from the original problem. All the routing decisions in constraints (2)-(12) are according to the current route given from one of the k-best routes found by the procedure in Section 4.2.2. This still leaves a significant number of integral priority variables. Our approach for reducing it is first to iteratively schedule one train at a time based on the order algorithm in Section 4.2.1. Then we hold the integrality for the priority variables associated with train $q^*$for one node at a time, and solve the relaxed problem. Thus, we solve **Subproblem-2** $|Z_{q^*}|$ times. Let i be the current node in Z_{q^*} that is being evaluated, we require integrality for $x_{q',q^*,i}$ and $x_{q^*,q',i}$ for $q' \in Q^{\Phi}$, and relax $x_{q',q^*,j}$ and $x_{q^*,q',j}$ for all $j \in Z_{q^*} \backslash \{i\}$. We also hold all the other priority variables fixed at their values from the previous iterations. Then to find integral values for the relaxed priority variables $x_{q',q^*,j}$ and $x_{q^*,q',j}$ for all $j \in Z_{q^*} \backslash \{i\}$, we apply a Backward-Forward Insertion (BFI) algorithm to derive their priorities.

On each node j in Z_{q^*}, we use S_j to represent all the previously scheduled trains that travel through node j, and an *insertion position* of q^* on node j is denoted as $a_j^{q^*}, 1 \leq a_j^{q^*} \leq |S_j| + 1$. Generally there are $|S_j| + 1$ candidate insertion positions on node j if there are $|S_j|$ previously scheduled trains passing through node j. Note that not all the sequences are legal, since some of the overtaking actions between trains are not allowed. First we define the *illegal insertion positions*. On each node j in Z_{q^*}, the insertion position $a_j^{q^*}$ separates S_j into two sets: $\hat{S}_{j,q^*}$ is the set of trains on node j that have higher priority than q^* , and $\check{S}_{j,q^*}$ is the set of trains on node j that have lower priority than q^*. For example in Figure 12 on node i if $a_i^{q^*} = 2$, then $\hat{S}_{i,q^*} = \{q_1\}$ and $\check{S}_{i,q^*} = \{q_2, q_3\}$.

Proposition 1: Any condition below yields an illegal insertion position.

a) $\check{S}_{i-1,q^*} \wedge \hat{S}_{i,q^*} \neq \emptyset, \forall i \in Z_{q^*}$

b) $\hat{S}_{i-1,q^*} \wedge \check{S}_{i,q^*} \neq \emptyset, \forall i \in Z_{q^*}$

c) $\tilde{t}^{d}_{q^*,q',i} + \sum_{k=i+1}^{j} B^1_{q^*,k} + \mu > t^{\mathrm{a}}_{q',j}, \forall i,j \in Z_{q^*}, i < j-1, \forall q' \in \hat{S}_{i,q^*} \cap \check{S}_{j,q^*}$

   $\tilde{t}^{\mathrm{d}}_{q^*,q',i}$ is the earliest departure time of q^* from node i after train q', $\tilde{t}^{\mathrm{d}}_{q^*,q',i} = t^{\mathrm{d}}_{q',i} + \mu + B^2_{q^*,i,i+1}$ and $t^{\mathrm{d}}_{q',i}$ is from the arrival and departure times recorded in the previous iterations.

Proof: Condition (a) and Condition (b) can be proved along similar approaches. We present the proof for Condition (a) by contradiction.

Assume $\check{S}_{i-1,q^*} \wedge \hat{S}_{i,q^*} \neq \emptyset, \forall i \in Z_{q^*}$, $\exists q' \in \check{S}_{i-1,q^*} \wedge \hat{S}_{i,q^*}$, $x_{q',q^*,i} = x_{q^*,q',i-1} = 1$. By Constraints (10),

$$\begin{cases} t^a_{q^*,i} \geq t^d_{q',i} + \mu \\ t^a_{q',i-1} \geq t^d_{q^*,i-1} + \mu \end{cases} \tag{26}$$

By Constraints (12), either $I_{q',i-1,i} = 1$ or $I_{q',i,i-1} = 1$.

(1) If $I_{q',i-1,i} = 1$, from Constraints (6), (7) and (9), we have

$$\left.\begin{aligned} t^a_{q,i} - t^a_{q,i-1} \geq B^1_{q,i} \Rightarrow t^a_{q',i} > t^a_{q',i-1} \\ t^d_{q',i} - t^d_{q',i-1} \geq B^1_{q',i} \;\Rightarrow\; t^d_{q',i} > t^d_{q',i-1} \\ t^d_{q',i-1} - t^a_{q',i} \geq B^2_{q,i-1,i} - \; B^1_{q,i} \Rightarrow t^d_{q',i-1} > t^a_{q',i} \end{aligned}\right\} \Rightarrow t^d_{q',i} > t^a_{q',i-1} \tag{27}$$

By Constraints (9),

$$t^d_{q^*,i-1} - t^a_{q^*,i} \geq B^2_{q^*,i-1,i} - \; B^1_{q^*,i} \Rightarrow t^d_{q^*,i-1} > t^a_{q^*,i} \tag{28}$$

Combining Inequalities (26) and (27) contradict with the combining of Inequalities (26) and (28) as follows

$$t^d_{q',i} > t^a_{q',i-1} > t^d_{q^*,i-1}$$

$$t^d_{q^*,i-1} > t^a_{q^*,i} > t^d_{q',i}$$

(2) If $I_{q',i,i-1} = 1$, by Constraints (9),

$$\begin{cases} t^d_{q^*,i-1} - t^a_{q^*,i} \geq B^2_{q^*,i-1,i} - \; B^1_{q^*,i} \\ t^d_{q',i} - t^a_{q',i-1} \geq B^2_{q',i,i-1} - \; B^1_{q',i} \end{cases} \Rightarrow \begin{cases} t^d_{q^*,i-1} > t^a_{q^*,i} \\ t^d_{q',i} > t^a_{q',i-1} \end{cases} \tag{29}$$

Condition (29) also contradicts with Condition (26).

Thus Condition (a) yields an illegal insertion position. Condition (b) can be proved along a similar approach.

For Condition (c), $\forall q' \in \hat{S}_{i,q^*} \cap \check{S}_{j,q^*}$, $x_{q',q^*,i} = \; x_{q^*,q',j} = 1$. By Constraints (10), we have

$$t^a_{q',j} \geq \; t^d_{q^*,j} + \; \mu \tag{30}$$

For the statement of Condition (c), we have

$$\tilde{t}^{\mathrm{d}}_{q^*,q',i} + \sum_{k=\,i+1}^{j} B^1_{q^*,k} + \; \mu > t^{\mathrm{a}}_{q',j} \tag{31}$$

And also since $\tilde{t}^d_{q^*,q',i}$ is the earliest departure time of q^* from node i after train q',

$$t^{\mathrm{d}}_{q^*,i} > \tilde{t}^{d}_{q^*,q',i} \tag{32}$$

From (31) and (32), we have

$$t^{\mathrm{d}}_{q^*,i} + \sum_{k=\,i+1}^{j} B^1_{q^*,k} + \,\mu > t^{\mathrm{a}}_{q',j} \tag{33}$$

By applying Constraints (7) on the sequence of nodes from $i+1$ to j, we have

$$t^{\mathrm{d}}_{q^*,j} \geq t^{\mathrm{d}}_{q^*,i} + \sum_{l=\,i+1}^{j} B^1_{q^*,l} \tag{34}$$

By substituting (32) to (31), we have

$$t^{\mathrm{d}}_{q^*,j} + \mu > t^{a}_{q',j}$$

which is a contradiction to Condition (30). Thus Condition (c) yields an illegal insertion position.

■

Proposition 1 states that on any consecutive nodes along the path, the priority relationship between any pair of trains cannot be swapped. And on any non-consecutive node pairs, the overtaking between q^* and q' can only happen if the free flow travel time of train q^* is short enough to leave node j earlier than the arrival of train q' within a safety headway. Note that in the proof of **Proposition 1**, q' does not need to travel in the same direction as q^*. Thus **Proposition 1** gives a dependent relationship of the insertion positions along the route, and we will use these dependencies to derive the priorities for train q^*. We define the *legal insertion position* as the position which does not create an illegal insertion position sequence.

Freight Insertion Rule (FIR): Among all the legal insertion positions, the front most position is selected to insert the newly scheduled train.

Under FIR, we select the nearest legal insertion position in order to reduce the waiting time of q^*. To minimize the travel time of q^*, the waiting time at the current node should be minimized. Thus the front most position should be preferred since it gives the minimal waiting time for q^* before the next node.

Based on **Proposition 1** and **FIR**, we propose our *Backward-Forward Insertion Algorithm* (BFI) to construct an insertion position sequence in an iterative approach. Recall, we currently have the priority decisions for node i for train q^* from the solution of **Subproblem-2**. BFI is used to infer the position of train q^* on all the other nodes in $Z_{q^*}\backslash\{\text{i}\}$. We use the conclusion from **Proposition 1** to eliminate the illegal insertion positions, and the conclusion from **FIR** to select the best legal insertion position. The algorithm performs a forward and a backward insertion position inference to determine the insertion position. Let $Z_{q^*}^{\Phi}$ be the nodes in which the priorities are known. Initially, $Z_{q^*}^{\Phi} = \{i\}$ in which i is the fixed integrality node from **Subproblem-2**. Then the insertion position on some other node j is inferred and it is included in $Z_{q^*}^{\Phi}$. The iterative approach is repeated until $Z_{q^*}\backslash Z_{q^*}^{\Phi}$ is empty. Note that the initial $\{i\}$ can be any node in Z_{q^*}. We use each of the nodes in Z_{q^*} for an initial solution, and the BFI algorithm is applied for each of the initial solutions to generate $|Z_{q^*}|$ final solutions. The best one that minimizes the objective of **Subproblem-2** is selected as the priority assignment for q^*.

We now present the details of the BFI algorithm. Starting from an initial insertion position, we use heuristic rules to infer the insertion positions on the rest of the $\left|Z_{q^*}\right| - 1$ nodes for train q^*. The idea of the BFI algorithm is that the existing insertion positions actually bounds the time window for the legal candidate insertion positions on the other nodes. Given the legal candidate insertion positions of a new node, the one that minimizes the objective function of **Subproblem-2** is selected. The backward-forward inference step contains two independent and similar parts, which are backward inference and forward inference. Each part is an iterative approach that uses the existing insertion positions to infer the insertion position on a new node.

The inference for a new node i' is based on a *feasible time window* on node i'. The feasible time window consists of four time measurements, defined as follows.

(a) $w_{q^*,i'}^{minArr}$: The earliest arrival time of train q^* to node i'.

(b) $w_{q^*,i'}^{maxArr}$: The latest arrival time of train q^* to node i'.

(c) $w_{q^*,i'}^{minDep}$: The earliest departure time of train q^* from node i'.

(d) $w_{q^*,i'}^{maxDep}$: The latest departure time of train q^* from node i'.

A feasible time window is calculated from the known insertion position(s) of the nodes. The algorithm is an iterative approach. In each iteration, the backward (forward) inference algorithm infers a new priority insertion position $a_j^{q^*}, j \in Z_{q^*} \backslash Z_{q^*}^{\Phi}$ and adds it to $Z_{q^*}^{\Phi}$, until all the nodes in Z_{q^*} are covered. We present the details of the backward inference and forward inference steps as follows.

Backward inference step:

Step_0: Starting from an insertion position $a_{i^*}^{q^*}$ on some node i^*, which is solved from **Subproblem-2**. Initialize $Z_{q^*}^{\Phi} = \{i^*\}$ and go to **Step_1**.

Step_1: Select node i', which is the previous node of i^* in Z_{q^*} to infer its insertion position. If there is no more previous node i' on route, go to **Step_4**. For each existing insertion positions on nodes $\{j \in Z_{q^*}^{\Phi}\}$, calculate the feasible time window on node i' with respect to j, denote as $w_{q^*,i'}^{minArr}(j)$, $w_{q^*,i'}^{maxArr}(j)$, $w_{q^*,i'}^{minDep}(j)$ and $w_{q^*,i'}^{maxDep}(j)$. Note that if $a_j^{q^*} = 1$, q^* is the first train passing node j, then $w_{q^*,i'}^{minArr}(j)$= 0; if $a_j^{q^*} = |S_j|$, q^* is the last train passing node j, then $w_{q^*,i'}^{maxArr}(j) = T^E$, which is the end of the day. Among the trains in S_j, we denote the train before $a_j^{q^*}$as $\hat{q}$ and the train after $a_j^{q^*}$as $\check{q}$, if they exist. We define the earliness of passenger train $q \in Q_p$ on node i as $e_{q,i} = \max(T_{q,s} - t_{q,s}^a, 0)$, where $i \in Z_q$ and s is the node that contains train q's next station stop after i. Note here earliness is defined differently from **Subproblem-1**, we cap the earliness at zero so it cannot go negative.

$$w_{q^*,i'}^{minArr}(j) = \begin{cases} 0 & a_j^{q^*} = 1, j \in Z_{q^*}^{\Phi} \\ \max\left(t_{\hat{q},j}^d + \mu - \sum_{l=i'+1}^{j} B_{q^*,l}^1 , 0 \right) & a_j^{q^*} > 1, j \in Z_{q^*}^{\Phi} \end{cases}$$

$$w_{q^*,i'}^{maxArr}(j)$$
$$= \begin{cases} \max\left(t_{\check{q},j}^{d} + \mu - \sum_{l=i'+1}^{j} B_{q^*,l}^{1}, 0\right) & a_j^{q^*} < |S_j| + 1 \ and\ \check{q} \in Q_f^{\Phi}, j \in Z_{q^*}^{\Phi} \\ \max\left(t_{\check{q},j}^{d} + \ e_{\check{q},j} + \ \mu - \sum_{l=i'+1}^{j} B_{q^*,l}^{1} \ ,0\right) & a_j^{q^*} < |S_j| + 1 \ and\ \check{q} \in Q_p, j \in Z_{q^*}^{\Phi} \\ T^E & a_j^{q^*} = |S_j| + 1, j \in Z_{q^*}^{\Phi} \end{cases}$$

$$w_{q^*,i'}^{minDep}(j) = \min(w_{q^*,i'}^{minArr}(j) + B_{q^*,i',i'+1}^{2}, T^E)$$

$$w_{q^*,i'}^{maxDep}(j) = \min(w_{q^*,i'}^{maxArr}(j) + B_{q^*,i',i'+1}^{2}, T^E)$$

The above calculations are directly derived from Constraints (6)-(11), which assume the free flow travel along the nodes in Z_{q^*}. Note that in the calculation for $w_{q^*,i'}^{maxArr}(j)$, if $\check{q}$ is a passenger train and will be early to its next station, then we relax the latest arrival time by the earliness time. By doing this, this passenger train which is early is then given a lower priority since it has slack in its schedule. Then the new scheduled freight train q^* could be scheduled before it. This step changes the departure time of the early passenger train, and is effective in reducing the freight train travel time while maintaining the minimum passenger tardiness.

Next, we identify the intersection of all time window measurements on node i' with respect to $j, j \in Z_{q^*}^{\Phi}$, to get the feasible time window on node i'. And then go to **Step 2**.

$$w_{q^*,i'}^{minArr} = \max_{j \in Z_{q^*}^{\Phi}} w_{q^*,i'}^{minArr}(j) \qquad w_{q^*,i'}^{maxArr} = \min_{j \in Z_{q^*}^{\Phi}} w_{q^*,i'}^{maxArr}(j)$$

$$w_{q^*,i'}^{minDep} = \max_{j \in Z_{q^*}^{\Phi}} w_{q^*,i'}^{minDep}(j) \qquad w_{q^*,i'}^{maxDep} = \min_{j \in Z_{q^*}^{\Phi}} w_{q^*,i'}^{maxDep}(j)$$

Step_2: We next identify the insertion position $a_{i'}^{q^*}$ from the above time windows. Rank trains in $S_{i'}$ from highest priority to lowest priority. If $|S_{i'}| > 1$, then remove train q' with the highest priority from $S_{i'}$, update $S_{i'} = S_{i'} \setminus \{q'\}$ and go to **Step_3**. Else if $|S_{i'}| = 0$, assign $a_{i'}^{q^*}$ to the end of the train list at i', update $Z_{q^*}^{\Phi} = Z_{q^*}^{\Phi} \cup \{i'\}$, and go to **Step_1**.

```
Step_3: For $q'$, retrieve arrival time $t^{a}_{q',i'}$ to node $i'$ and departure time $t^{d}_{q',i'}$ from node $i'$
    If $w^{maxDep}_{q^*,i'} + \mu < t^{a}_{q',i'}$ :
        Schedule $q^*$ before $q'$, $a^{q^*}_{i'} = a^{q'}_{i'}$. Update $Z^{\Phi}_{q^*} = Z^{\Phi}_{q^*} \cup \{i'\}$, and go to Step_1.
    Else if $t^{d}_{q',i'} + \mu > w^{maxArr}_{q^*,i'}$ and $w^{minDep}_{q^*,i'} + \mu < t^{a}_{q',i'}$:
        Schedule $q^*$ before $q'$, $a^{q^*}_{i'} = a^{q'}_{i'}$, update $Z^{\Phi}_{q^*} = Z^{\Phi}_{q^*} \cup \{i'\}$, and go to Step_1.
    Else if $w^{minArr}_{q^*,i'} > t^{d}_{q',i'} + \mu$:
        Evaluate next position in $S_{i'}$ and return to Step_2.
    Else if $w^{minDep}_{q^*,i'} + \mu > t^{a}_{q',i'}$ and $t^{d}_{q',i'} + \mu < w^{maxArr}_{q^*,i'}$:
        Evaluate next position in $S_{i'}$ and return to Step_2
    Else if $w^{minDep}_{q^*,i'} + \mu < t^{a}_{q',i'}$ and $w^{maxArr}_{q^*,i'} > t^{d}_{q',i'} + \mu$ :
        If $\frac{t^{a}_{q',i'} - (w^{minDep}_{q^*,i'} + \mu)}{w^{maxDep}_{q^*,i'} - w^{minDep}_{q^*,i'}} > \frac{w^{maxArr}_{q^*,i'} - (t^{d}_{q',i'} + \mu)}{w^{maxArr}_{q^*,i'} - w^{minArr}_{q^*,i'}}$:
            Schedule $q^*$ before $q'$, $a^{q^*}_{i'} = a^{q'}_{i'}$, update $Z^{\Phi}_{q^*} = Z^{\Phi}_{q^*} \cup \{i'\}$, and go to Step_1.
        Else: Evaluate next position in $S_{i'}$ and return to Step_2.
Step_4: Return the insertion positions $\{a^{q^*}_{1}, a^{q^*}_{2} ... a^{q^*}_{i^*}\}$
```

In Figure 13, the inference rules in **Step_3** are illustrated. The decision of scheduling q^* before q' is denoted as $q^* > q'$, and the decision of scheduling q^* after q' is denoted as $q^* < q'$ in the figure.

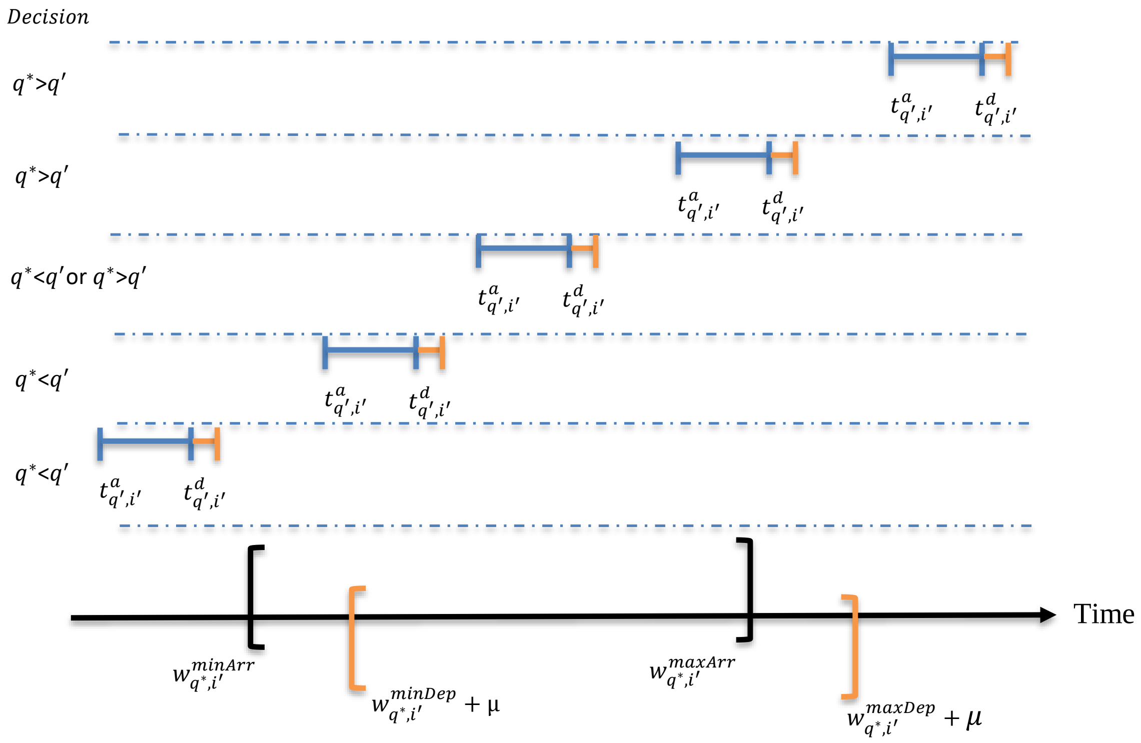


Figure 13. Inference Rules in **Step_3**

Forward inference step:

The forward inference algorithm follows the same logic as the backward inference algorithm. In **Step_1**, node i' is selected as the next node of i^* in Z_{q^*} in each iteration. In **Step_1**, the calculation for $w^{minDep}_{q^*,i'}(j)$ and $w^{maxDep}_{q^*,i'}(j)$ remains the same as the backward inference step, the calculation for $w^{minArr}_{q^*,i'}(j)$ and $w^{maxArr}_{q^*,i'}(j)$ changes to

$$w^{minArr}_{q^*,i'}(j) = \begin{cases} 0 & a^{q^*}_j = 1, j \in Z^{\Phi}_{q^*} \\ \min(t^d_{\hat{q},j} + \mu + \sum_{l=j+1}^{i'} B^1_{q^*,l}, T^E) & a^{q^*}_j > 1, j \in Z^{\Phi}_{q^*} \end{cases}$$

$$w_{q^*,i'}^{maxArr}(j) = \begin{cases} \min\left(t_{\check{q},j}^{d} + \mu + \sum_{l=j+1}^{i'} B_{q^*,l}^{1}, T^E \right) & a_j^{q^*} < |S_j| + 1 \; and \; \check{q} \in Q_f^{\Phi}, j \in Z_{q^*}^{\Phi} \\ \min\left(t_{\check{q},j}^{d} + e_{\check{q},j} + \mu + \sum_{l=j+1}^{i'} B_{q^*,l}^{1}, T^E \right) & a_j^{q^*} < |S_j| + 1 \; and \; \check{q} \in Q_p, j \in Z_{q^*}^{\Phi} \\ T^E & a_j^{q^*} = |S_j| + 1 \end{cases}$$

In our proposed algorithm, an insertion position sequence is constructed for each initial insertion position, thus at most $|Z_{q^*}|$ sequences are constructed. Then each of the sequences (priorities decisions) and routing decisions are substituted into **Subproblem-2** and a linear programming problem is solved for the departure and arrival time of all the trains. The sequence with the minimal objective value is finalized as the insertion position sequence for q^* on route Z_{q^*}, and the updated departure/arrival times of all the other trains are also obtained from the solution.

4.2.3.1 Freight Train Priority Assignment Experiments

The Backward-Forward Insertion (BFI) algorithm solves $|Z_{q^*}|$ integer problems. Recall each integer program maintains the integrality requirement for all the $x_{q',q^*,i}$ and $x_{q^*,q',i}$ variables for node i and $q' \in Q^{\Phi}$. All the remaining priority variables $x_{q',q^*,j}$ and $x_{q^*,q',j}$ where $j \in Z_{q^*} \backslash \{i\}$ in **Subproblem-2** are relaxed. Note that the priority variables between any two previously scheduled trains $q_1, q_2 \in Q^{\Phi}$ are fixed to their values from the previous iterations. Then after obtaining the $x_{q',q^*,i}$ and $x_{q^*,q',i}$ variables from solving **Subproblem-2**, we apply the BIF algorithm to obtain the priority variables $x_{q',q^*,j}$ and $x_{q^*,q',j}$ for all $j \in Z_{q^*} \backslash \{i\}$. An alternative approach to using the BFI algorithm is to require integrality for all the variables $x_{q',q^*,j}$ and $x_{q^*,q',j}$ for all $j \in Z_{q^*}$ and $q' \in Q^{\Phi}$. Note that the although the latter approach will only solve one integer program, it will take significantly more computation time since there are more integer variables in this formulation. We refer to this approach as **Full Priority Assignment** (FPA). We next perform the comparison between the BFI algorithm and FPA algorithm. This experiment is performed on the test network that was introduced in Section 4.2.1.1. We add 84 freight trains and 22 demand sets to the previous data sets. The two objectives in **Subproblem-2** are weighted with a coefficient of $\beta = 0.5$. This coefficient is also used for the rest of the experiments. For the k-best routes, we set $k = 1$. The experiments are conducted on a PC with a 3.6 GHz Intel Core CPU and 16GB memory. In summary, we first perform the procedure described in Section 4.1 to determine the passenger train priorities, routes and departure/arrival times. Then we apply the algorithm to determine the freight train scheduling order. Then we iteratively determine the routing, priority and departure/arrival time for each freight train. We use the algorithm described in Section 4.2.2 to determine the routes, and we use the BFI algorithm or the FPA algorithm to determine the freight train priorities. Note that although at each iteration, the previously scheduled train routes and priorities are fixed, their departure/arrival times maybe adjusted when scheduling the new freight train q^*. After the BFI algorithm, a linear program is solved to get the new departure/arrival times for all the trains.

In the results shown in Table 4, each row is the result of scheduling one additional freight train, and each row depends on the priority sequences of the previously scheduled trains. Table 4 shows the comparison of the objective function and CPU time between the two algorithms.

	FPA		BFI			FPA		BFI			FPA		BFI	
Index	Objective	CPU (s)	Objective	CPU (s)	Index cont.	Objective	CPU (s)	Objective	CPU (s)	Index (cont.)	Objective	CPU (s)	Objective	CPU (s)
1	40.5	0.7	40.5	0.3	30	1039.4	18.1	1045.7	2.7	59	1937.4	31.2	1940.3	15.9
2	79.7	0.8	79.7	0.4	31	1071.9	7.5	1083.3	3.4	60	1975.1	32.3	1971.4	13.0
3	116.9	0.9	116.9	0.4	32	1107.4	8.4	1114.9	3.6	61	2005.0	33.1	2002.5	13.5
4	150.7	1.1	150.7	0.4	33	1139.9	8.6	1146.1	3.3	62	2038.7	35.9	2033.6	17.7
5	185.5	1.0	185.5	0.5	34	1172.0	8.7	1178.3	4.4	63	2071.1	35.2	2066.5	12.7
6	217.5	1.1	218.3	0.5	35	1202.5	10.3	1209.9	6.7	64	2108.7	40.2	2098.0	16.5
7	253.5	1.3	253.5	0.6	36	1235.8	10.1	1240.4	5.8	65	2131.2	41.2	2119.7	10.8
8	291.4	1.4	291.4	0.7	37	1266.1	14.1	1273.3	2.7	66	2151.6	39.3	2142.2	9.2
9	326.7	4.6	332.3	0.7	38	1296.8	6.2	1303.3	10.3	67	2192.6	38.5	2175.0	19.5
10	365.2	2.3	365.3	0.8	39	1328.0	12.4	1335.6	7.4	68	2225.0	45.3	2206.9	17.6
11	397.7	3.9	398.1	0.9	40	1365.8	9.7	1372.7	6.8	69	2260.5	46.3	2245.6	17.5
12	430.3	7.8	430.8	0.9	41	1398.4	13.8	1404.6	4.7	70	2292.0	47.0	2277.7	14.2
13	465.5	13.1	466.0	1.2	42	1428.9	13.8	1438.0	5.4	71	2329.7	48.6	2309.1	14.0
14	495.9	7.4	497.6	1.0	43	1450.9	9.3	1469.5	6.3	72	2361.3	49.3	2347.0	13.5
15	531.5	3.0	533.2	1.5	44	1472.6	10.8	1501.7	6.7	73	2395.2	50.8	2380.3	13.2
16	567.2	3.6	569.1	1.5	45	1501.5	7.7	1532.9	4.6	74	2433.1	51.3	2414.2	18.2
17	600.6	4.1	602.3	1.7	46	1523.3	10.2	1560.1	4.1	75	2467.7	54.3	2447.6	18.1
18	635.0	3.0	637.1	1.7	47	1543.1	10.5	1581.6	5.2	76	2499.7	52.4	2479.6	18.1
19	668.1	9.2	677.1	1.9	48	1562.9	15.7	1607.4	6.4	77	2532.7	51.3	2510.4	17.3
20	706.7	5.5	707.9	2.6	49	1602.4	8.6	1625.1	4.8	78	2567.4	50.3	2541.4	18.0
21	739.9	5.4	741.1	3.1	50	1635.5	10.4	1645.5	5.6	79	2602.0	53.5	2575.2	17.4
22	773.0	6.8	774.2	3.6	51	1669.5	6.4	1667.6	7.2	80	2633.5	54.9	2607.8	17.0
23	806.2	15.1	808.1	3.3	52	1703.6	10.7	1708.4	7.2	81	2668.6	55.5	2641.6	21.3
24	840.2	12.5	841.4	3.5	53	1736.9	19.3	1740.5	13.6	82	2705.5	56.8	2674.2	13.8
25	874.6	11.1	875.6	3.3	54	1770.9	20.1	1772.7	16.6	83	2736.0	58.5	2705.4	15.1
26	907.3	13.0	908.8	4.5	55	1803.4	20.0	1806.2	11.8	84	2766.5	58.8	2736.4	23.1
27	940.1	10.5	941.4	2.2	56	1837.9	17.0	1840.7	15.8					
28	976.6	18.1	975.6	2.7	57	1874.5	22.6	1874.5	17.8					
29	1008.1	7.5	1008.8	3.4	58	1907.5	14.4	1907.7	30.1					

Table 4. Objective Comparison Between BFIPC and Opt-Integer with Same Passenger Schedule

Note that both algorithms use an iterative approach, since the selected insertion position sequence $\{a_i^{q^*}, i \in Z_{q^*}\}$ is fed to the next iteration of the next train to schedule, so that both algorithms give sub-optimal results. The results from the test network shows that the solution quality for the BFI and FPA algorithms are similar. Since the priorities for the previous iterations are fixed in solving the schedule of the additional train, the BFI can sometimes even outperform the FPA algorithm. From the comparison of CPU time, BFI is much faster than the FPA algorithm, especially when the number of freight train increases without much loss in solution quality.

5. Experimental Results

In this section, we compare the performance of the proposed algorithms with other solution methods. In Section 5.1, we compare our solution approach against the optimal solutions. This comparison can only be made for small rail networks since it is computationally difficult to find optimal solution for large networks. In Section 5.2, we compare our solution with other heuristic methods for a large rail network.

5.1. Small Network

First we build a small rail network with a small number of trains. The original model is directly solved on this sample network and passenger train timetables using a commercial optimization software, CPLEX. Then we deploy our algorithm to solve the same model and compare the objective and CPU time. Here, we use the weighted objective function with β=0.5.

The sample network contains 16 miles of rail track with double and triple track segments. There are six junctions along the track and three train stations in the rail network, and the abstract graph contains 49 nodes and 63 arcs. There are a total of 5 passenger trains. The railway trackage is show in Figure 14 and the timetable for the two schedules are shown in Table 5. In the Uniform-schedule, the passenger train timetable is uniformly distributed throughout the day. In the Compact-schedule, the passenger train timetable is scheduled within two rush hour periods in 8:00-10:00 and 17:00-19:00.

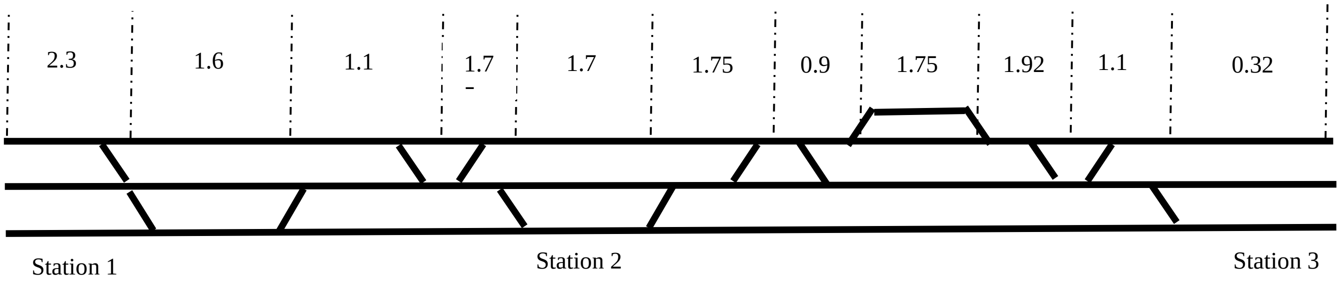


Figure 14. Test Network with Three Stations

Passenger Train Index	Uniform-Schedule			Compact-Schedule		
	Station 1	Station 2	Station 3	Station 1	Station 2	Station 3
1	00:00	00:16	00:41	09:09	09:25	09:50
2	05:00	05:17	05:35	09:25	09:42	10:00
3	10:24	10:10	10:00	17:24	17:10	17:00
4	15:29	15:15	15:00	17:49	17:35	17:25
5	20:24	20:10	20:00	17:54	17:40	17:30

Table 5. Test Passenger Train Schedule

In the mathematical model, there are 2,538 integer variables. First we directly solve the model using CPLEX, and denote the result as the CPLEX-Solution. Then we deploy our heuristic algorithm to solve the same model, and denote the result as Heuristic-Solution. The results are presented in Table 6. Note that we stop CPLEX when either the optimal solution is found, or the maximum CPU time of 10,000s is reached. If the optimal solution is not found, we record the best solution found by CPLEX. The Opt Gap column gives the optimality gap for the CPLEX solution. From the results, our Heuristic-Algorithm computes the solution within 60s CPU time for all schedules. Note that when freight train number is greater than seven, the optimality gap is very large, most likely due to a poor lower bound found by Cplex. Our heuristic solution is very close to the optimal solution (or best CPLEX solution) in all schedules. Thus we conclude our solution approach can find near optimal solutions for this small rail network when it is known.

# of Freight Trains	Uniform-Schedule					Compact-Schedule				
	CPLEX-Solution			Heuristic-Solution		CPLEX -Solution			Heuristic-Solution	
	Objective	Opt Gap (%)	CPU time (s)	Objective	CPU time (s)	Objective	Opt Gap (%)	CPU time (s)	Objective	CPU time (s)
4	49.117	0.0	40.1	49.222	39.2	49.104	0.0	57.1	50.518	42.5
5	60.156	0.0	413.8	65.479	40.5	59.343	0.0	695.1	61.459	40.9
6	73.657	0.0	7,508	74.530	42.5	73.657	0.0	7,692	73.828	43.9
7	83.950	2,847	10,000	84.769	46.0	84.76	3510	10,000	84.769	46.2
8	99.135	3,531	10,000	99.136	48.3	98.317	4240	10,000	102.312	48.4

Table 6. Comparison Between Optimal-solution and Heuristic-Solution

5.2. Large network

In Section 4, we illustrate the details of our proposed algorithm and tune the parameters of the algorithm based on its performance for a medium size rail network. In Section 5.1, we compare the overall CPU time between our solution approach and the optimal solution on a solvable small network, and we discuss the solution quality. Now we move to a large scale complex network to test the performance of our algorithm.

We consider a 59 miles long rail track network from Los Angeles to Riverside, CA. The trackage configuration consists of double-track segments and triple-track segments, and it contains eight passenger stations. In this network, there are 266 track segments and 88 junctions, and our translated abstract graph contains 331 nodes and 319 arcs. We define a base case schedule according to the daily schedule of two passenger rail service providers in this area, Amtrak and Metrolink. In the base case, there are a total of 84 freight trains and 89 passenger trains per day.

To schedule the trains movement in the network, one approach is to follow our solution approach to get the decision variables, including routing decisions, priority decisions and departure/arrival time decisions. This approach uses our solution approach to control these train movements according to these decisions. We name this approach as *Complete-Control*. However, it may be difficult to strictly follow the *Complete-Control* solution in practice due to real-time changes and unexpected events in rail operations. Another approach is to deploy the departure/arrival time decisions from our solution model, and use a greedy based algorithm to construct the routes and assign the priorities as the trains travel along the rail network, similar to the approach of Lu and Dessouky (2004). The greedy based algorithm assigns priorities to trains under a FCFS rule. The routing decision is made when a train approaches the end of the node. If there are multiple available successor nodes, the node that gives the best performance for the system is selected. The performance is evaluated by a depth-first-search-like algorithm. If there is no available successor node, the train decelerates to prepare for stopping at the end of the node. We name this approach as *Partial-Control*. The third approach is to randomly determine the freight train departure times and uses the greedy algorithm for routing and priority assignment, which is referred as *Random-Departure*. The fourth approach is to assign the freight trains with uniform departure times (equal interval) and uses the greedy algorithm for routing and priority assignment, which is referred as *Uniform-Departure*. In both the *Random-Departure* and *Uniform-Departure* approaches, the departure times of passenger trains from the origin station follow the timetable.

A comparison of the four approaches is presented in Table 7. Freight train delay is the difference between the actual travel time and the free flow travel time. In the base case schedule, there are 84 freight trains per day, and there are 89 passenger trains with 260 passenger train and station combinations. Passenger train tardiness percentage is the percentage of times that a passenger train arrived tardy to a station and passenger train average tardiness is the average tardiness of these tardy arrivals. We increase the number of freight trains by 11 each time, and evaluate the performance of our approach when freight train demand increases.

# of Freight Train	Complete-Control			Partial-Control		
	Freight train	Passenger train		Freight train	Passenger train	
	Average Delay (min)	Tardiness Percentage (%)	Average Tardiness (min)	Average Delay (min)	Tardiness Percentage (%)	Average Tardiness (min)
84	1.634	0	0	3.282	5.96	1.158
95	1.701	0	0	3.484	7.78	1.259
106	1.752	0	0	3.761	8.52	1.261
117	1.983	0	0	4.733	8.96	1.272
128	2.047	0	0	5.076	9.88	1.387
139	2.122	0	0	6.812	10.00	1.490
150	2.363	0	0	8.944	10.78	1.495

# of Freight Train	Random-Departure			Uniform-Departure		
	Freight train	Passenger train		Freight train	Passenger train	
	Average Delay (min)	Tardiness Percentage (%)	Average Tardiness (min)	Average Delay (min)	Tardiness Percentage (%)	Average Tardiness (min)
84	8.223	28.06	5.440	8.029	27.59	5.517
95	9.404	31.74	6.036	10.361	30.98	6.101
106	10.121	34.82	7.745	11.405	35.71	7.697
117	11.332	38.64	8.046	11.809	39.02	7.912
128	12.126	39.79	8.969	12.837	40.11	8.871
139	13.589	44.28	11.244	13.522	44.03	11.296
150	14.262	48.58	11.654	14.901	49.01	11.493

Table 7. Performance Comparison Between Complete-Control, Partial-Control, Random-Departure and Uniform-Departure

In Table 7, the *Complete-Control* control approach outperforms the other three approaches in reducing the freight train average delay and the passenger train tardiness. The *Random-*

Departure and *Uniform-Departure* have similar results, since the departure times of the freight trains do not consider the traffic congestion of the network. Thus the delay of the freight trains and tardiness of the passenger trains grows with increasing number of freight trains. Note that the average free flow travel time of freight trains is about 72 minutes in all cases, so when the number of freight trains increases, the delay of the freight trains actually contributes to a larger percentage to their total travel time. The results from *Complete-Control* and *Partial-Control* have smaller freight average delay time and average passenger train tardiness, since the departure time decision is optimized according to the traffic in the rail network. Among the four approaches, the *Complete-Control* approach gives zero average passenger train tardiness and the smallest freight train delay. This indicates that to meet the expanded freight train capacity, our proposed solution procedure provides an efficient and high quality solution for the integrated passenger and freight train scheduling and routing problem.

6. Implementation

This project addresses the problem of integrating passenger and freight rail scheduling. Rail transportation is an efficient way to move commodities and passengers across the country. The expected rapid growth in freight rail transportation will introduce additional congestion to the current rail network and influence the passenger trains' schedule. Methodologies that can improve the scheduling of trains will improve the overall efficiency of the logistics system and expand the capacity of the railway network with minimal disruption to passenger rail service.

The integrated routing and scheduling of freight and passenger train model, and the proposed decomposition heuristics developed as part of this funded research project were tested on actual data for the Los Angeles rail network. The computational experiments show that our approach compares favorably against other approaches. The implementation of our heuristics will require the programming language Python and an optimization solver, such as CPLEX. It also requires converting an actual rail network into an abstract graph as described in Section 3.1. The entire solution framework including all heuristics introduced is implemented in Python.

7. Conclusions

In the United States, rail transportation offers a viable mode to transport freight and passengers across the country. Due to the rapid increase in international trade, there has been a significant increase in railway traffic across the nation. In order for rail transportation to continue to be efficient, an in-depth understanding and study of how congestion occurs in a railway network, how to prevent it and how to reduce the delay experienced by trains is required. While there exists prior work in the areas of railway routing and scheduling, there has been little work that integrates the scheduling of passenger and freight trains to minimize passenger train tardiness and freight train travel times respectively, while accounting for the complexity and scalability of real-world rail operations. To address this gap, we present a solution approach involving an integrated routing and scheduling model that can be used to optimize the travel times of the freight trains and tardiness of the passenger trains. To solve the problem for real-world size problems, a vertical decomposition is first performed for passenger train schedule optimization and then combined with freight train scheduling in an iterative procedure.

From the results presented in Section 4.2.3.1, our decomposition based solution framework solves the problems more efficiently than directly deploying a commercial optimization solver such as CPLEX, while maintaining a high quality solution. In Section 5, we first show that our decomposition heuristic procedure provides solutions that are near optimal on a test problem for a small rail network. We can only compare to the optimal solution for small size problems, since it is computationally hard to find optimal solutions for large rail networks. Thus, for a large rail network, we compare our decomposition procedure with other heuristic approaches, and results show that our heuristic outperforms other heuristic rules.

8. References

Association of American Railroads. 2014. Overview of America's Freight Railroads.

Brännlund, U., Lindberg, P.O., Nõu, A. and Nilsson, J.E., 1998. Railway timetabling using Lagrangian relaxation. *Transportation science*, *32*(4), pp.358-369.

Cacchiani, V., Caprara, A. and Toth, P., 2010. Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological*, *44*(2), pp.215-231.

Caprara, A., Fischetti, M. and Toth, P., 2002. Modeling and solving the train timetabling problem. *Operations research*, *50*(5), pp.851-861.

Caprara, A., Monaci, M., Toth, P. and Guida, P.L., 2006. A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete applied mathematics*, *154*(5), pp.738-753.

Carey, M. and Lockwood, D., 1995. A model, algorithms and strategy for train pathing. *Journal of the Operational Research Society*, pp.988-1005.

Cordeau, J.F., Toth, P. and Vigo, D., 1998. A survey of optimization models for train routing and scheduling. *Transportation science*, *32*(4), pp.380-404.

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

D'ariano, A., Pacciarelli, D. and Pranzo, M., 2007. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, *183*(2), pp.643-657.

Dessouky, M.M., Lu, Q., Zhao, J. and Leachman, R.C., 2006. An exact solution procedure to determine the optimal dispatching times for complex rail networks. *IIE transactions*, *38*(2), pp.141-152.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numerische mathematik, 1(1), pp.269-271.

Dorfman, M.J. and Medanic, J., 2004. Scheduling trains on a railway network using a discrete event model of railway traffic. Transportation Research Part B: Methodological, 38(1), pp.81-98.

Eppstein, D., 1998. Finding the k shortest paths. *SIAM Journal on computing*, *28*(2), pp.652-673.

Harrod, S.S., 2012. A tutorial on fundamental model structures for railway timetable optimization. *Surveys in Operations Research and Management Science*, *17*(2), pp.85-96.

Higgins, A., Kozan, E. and Ferreira, L., 1996. Optimal scheduling of trains on a single line track. *Transportation research part B: Methodological*, *30*(2), pp.147-161.

Jovanović, D. and Harker, P.T., 1991. Tactical scheduling of rail operations: the SCAN I system. *Transportation Science*, *25*(1), pp.46-64.

Louwerse, I. and Huisman, D., 2014. Adjusting a railway timetable in case of partial or complete blockades. *European Journal of Operational Research*,*235*(3), pp.583-593.

Lu, Q., Dessouky, M. and Leachman, R.C., 2004. Modeling train movements through complex rail networks. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *14*(1), pp.48-75.

Lusby, R.M., Larsen, J., Ehrgott, M. and Ryan, D., 2011. Railway track allocation: models and methods. *OR spectrum*, *33*(4), pp.843-883.

Mascis, A. and Pacciarelli, D., 2002. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, *143*(3), pp.498-517.

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

Meng, L. and Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables. *Transportation Research Part B: Methodological*, *67*, pp.208-234.

Motraghi, A. and Marinov, M.V., 2012. Analysis of urban freight by rail using event based simulation. Simulation Modelling Practice and Theory, 25, pp.73-89.

Mu, S. and Dessouky, M., 2011. Scheduling freight trains traveling on complex networks. *Transportation Research Part B: Methodological*, *45*(7), pp.1103-1123.

Higgins, A., Kozan, E. and Ferreira, L., 1996. Optimal scheduling of trains on a single line track. *Transportation research part B: Methodological*, *30*(2), pp.147-161.

Zhou, X. and Zhong, M., 2007. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds.*Transportation Research Part B: Methodological*, *41*(3), pp.320-341.