

A decomposition based hybrid heuristic algorithm for the joint passenger and freight train scheduling problem

Liang Liu and Maged Dessouky

Daniel J. Epstein Department of Industrial & Systems Engineering

University of Southern California

Los Angeles, California 90089

maged@usc.edu

(213) 740-4891

Abstract

We study the joint problem of scheduling passenger and freight trains for complex railway networks, where the objective is to minimize the tardiness of passenger trains at station stops and the delay of freight trains. We model the problem as a mixed integer program and propose a two-step decomposition heuristic to solve the problem. The heuristic first vertically decomposes the train schedules into a passenger train scheduling phase and then a freight train scheduling phase. In the freight train scheduling phase, we use a train-based decomposition to iteratively schedule each freight train. Experimental results show the efficiency and quality of the proposed heuristic algorithm on real world size problems.

Keywords

Train routing, scheduling, heuristics

1. Introduction

In the United States, the rail industry has been at the forefront of economic growth. Both the freight and passenger train demand have been increasing over the years. Given that the freight industry is already running without much excess capacity, the rail industry has to either expand or manage its operations more efficiently. Investing in the expansion of the railroad network is an expensive venture. Thus, better scheduling methodologies become an effective solution to the problems caused by increasing transportation demand under tight capacity constraints. However, passenger and freight train scheduling are rather different. Whereas passenger train schedules are relatively stable and periodic, 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 passenger and freight 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 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 joint 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 jointly perform passenger and freight scheduling.

In this paper, we consider the daily problem of scheduling both passenger and freight trains jointly. The model assumes that only the scheduled departure times of the passenger trains from their stations are given and the arrival times at the all track segments including track segments with stations are decision variables for both the passenger and freight trains to minimize the bi-objective of passenger train tardiness (difference of the actual departure time and scheduled departure time at the passenger train stations) and freight train travel delay. We note there has been

some limited research that considers both passenger and freight trains or trains of different priority class in determining the schedule. Those prior papers treat the timetables for the passenger trains (or trains of higher class) at all track segments as fixed and schedule the freight trains (or trains of lower priority) around these timestamps. Different than the prior research is that (1) the passenger timetables at the stations are only given and the model determines the arrival time decisions at all other track segments such as at the junctions, control signals, etc., and (2) the passenger timetables at the stations are treated as soft constraints and a penalty is applied to the schedule when the actual departure time deviates from it. Thus, we formulate a model to jointly perform *daily* freight and passenger train scheduling and propose an algorithm that can schedule trains efficiently to minimize passenger train tardiness and freight train delay. Although the passenger train timetables at the stations are stable, we are concerned with daily schedules since the demand for freight trains can change on a daily basis and this freight demand could have an impact in the actual arrival times of the passenger trains at non-station track segments. We assume the freight train demand is known before the daily planning horizon. The remainder of this paper is organized as follows. A literature review on both train scheduling and timetabling is presented section 2. In Section 3, we formally define 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. Finally, conclusions are presented in Section 6.

2. Literature Review

A considerable amount of literature has been published on the rail scheduling problem. Cordeau et al. (1998), Caprara et al. (2007), Lusby et al. (2011) and Harrod (2012) surveyed the optimization models for the train routing, scheduling and timetabling problem. A recent survey by Hansen (2010), summarized emerging methods and the key solution approaches for the train timetabling and dispatching problem.

Since the railway scheduling and timetabling problems often overlap, we first start with the review of the railway scheduling problem and then the train timetabling problem. A significant amount of the train scheduling related literature study special problems arising in the operation of large railway networks. For example, several variations of the single-line track train scheduling problems are investigated. Oliveira and Smith (2000) modelled the single-track railway scheduling problem as a special case of the job-shop scheduling problem with the objective of minimum flow

time. Conflicts occur when the desired timetable results in two or more trains occupying the same track at the same time. Caprara et. al (2002) concentrated on the problem of a single, one-way track linking two major stations, with a number of intermediate stations in between. Caprara et. al (2006) extended the aforementioned model by considering several additional constraints for real-world applications. Furthermore, Cacchiani et. al (2008) proposed heuristic and exact algorithms of the same problem based on the solution of the LP relaxation of an ILP formulation, in which the decision variables correspond to a full timetable for a train. Zhou and Zhong (2007) proposed a generalized resource-constrained project scheduling formulation for a single-line track network. Bersani et al. (2015) addressed the problem of scheduling trains on a single track from a robust optimization point of view.

Another trend of research in railway operations is the dispatching and scheduling for multiple tracks or N-tracks in a complex rail network. Dessouky et. al (2006) developed a mathematical programming model to determine the optimal dispatching times for complex rail networks in densely populated metropolitan areas. Mazzarello and Ottaviani (2007) deployed an alternative graph model, which was first introduced by Mascis and Pacciarelli (2002), to the implementation of a Traffic Management System. Liu and Kozan (2009) modeled the train scheduling problem as a blocking parallel-machine job shop scheduling problem. The model is based on the disjunctive graph formulation by Roy and Sussman (1964) and is solved by an improved shifting bottleneck procedure algorithm and a feasibility satisfaction procedure algorithm. In Corman et al. (2011), an innovative optimization framework is proposed for the multi-class rescheduling problem when perturbation occurs in the network. The framework provides high quality schedules when combined with an advanced branch and bound algorithm in D'Ariano et al. (2007). In the sequential optimization framework, only one priority class is considered in each iteration. Meng and Zhou (2011) studied the method to recover the impacted train schedule from the current and future disturbances to minimize the expected additional delay under different forecasted operational conditions. Mu and Dessouky (2011) considered the freight train scheduling problem for a complex railway network which considers flexible routing. A freight train routing and scheduling problem is formulated as a mixed integer programming model, based on binary variables to explicitly model the train movement sequence and priorities. Meng and Zhou (2014) proposed a solution procedure to optimize N-track train schedules through simultaneous rerouting and rescheduling. Sama et al. (2015) studied a filtering method for the

routing selection and scheduling problem, and an ant colony optimization method is proposed in the solution procedure. Samà et al. (2016) studied the real-time train routing selection problem by revisiting the real-time Railway Traffic Management Problem (rtRTMP) and then introduced a new real-time Train Routing Selection Problem (rtTRSP) formulation. Šemrov et al. (2016) introduced a train rescheduling method based on reinforcement learning theory and solved for the scheduling problem of both single-track and complex railway networks.

We now move to a brief review on the train timetabling problem on a railway network. Railway timetables specify the arrival and departure times of trains at stations, yards, sidings and other check points. The operation of passenger trains is highly dependent on these timetables. Numerous studies have focused on the Train Timetable Problem (TTP). For example, Brannlund et al. (1998) presented an optimization approach for the timetabling problem in which the single track problem with sidings is modeled as an integer problem. Semet and Schoenauer (2005) described an evolutionary algorithm that relies on a semi-greedy heuristic to reconstruct the schedule by inserting trains one after the other following the permutation, with the aim of minimizing the total delay. Liebchen and Mohring (2007) proposed the decision for the integration of network planning, line planning and vehicle scheduling in the task of periodic timetabling. Flier et al. (2009) studied adding an additional train to the timetable on a corridor with minimum possible delay. Cacchiani et al. (2010) studied the freight train scheduling problem with a prescribed passenger train timetable. The timetable of passenger trains cannot be changed while new freight trains are inserted into the system. Prioritized and non-prioritized trains in railway scheduling are addressed in Liu and Kozan (2011). In Sun et al. (2014), a multi-objective optimization model is proposed to minimize the deviation for train rerouting on a high-speed railway network. Shafia et al. (2012) proposed a robust timetabling model for a single track railway line to compute buffer times. Louwse and Huisman (2014) studied the adjusted timetable of passenger trains in case of major disruptions. The problem is formulated using an event-activity network in Schachtebeck and Schöbel (2010). In Veelenturf et al. (2016), a macroscopic level timetable rescheduling problem is studied for passenger trains. The number of cancelled and delayed trains is minimized under infrastructure and rolling stock capacity constraints. Lamorgese and Mannino (2016) studied the real time rescheduling and rerouting problem in an attempt to minimize the deviations from the given timetable, and solve the mixed integer linear programming model with a Benders-like decomposition. Several other related studies can be found in Arenas et

al. (2015), Karoonsoontawong et al. (2015), Goverde et al. (2016) and Bešinović et al. (2016). In Cacchiani et al. (2016), a train timetabling problem is studied to determine the best schedule for a given set of trains under the constraints of track operational constraints. Ideal timetable and the prescribed route for each train are given and overtaking is only allowed within stations. Compare with this research, we allow for the possibility of overtaking at each junction, which makes the problem structure more complex.

Most of the literature in the domain of passenger train timetabling identifies the arrival and departure times of trains at stations, yards, sidings and other checkpoints. In our problem, we assume the scheduled departure times only at the stations are given and the arrival times at the other track segments are decision variables to minimize the bi-objective of passenger train tardiness and freight train travel delay. To accurately model the movement of each train traveling on each track segment we use a microscopic modelling approach where each node (track segment) represents the length of the minimum headway between the trains; thus, the time decisions are made at the microscopic level. There has been some literature using microscopic level modelling. For example, Corman et al. (2011) developed a microscopic level model of railway traffic to reschedule the prioritized trains when conflicts occur. In this paper, we study the joint scheduling of passenger and freight trains, where both passenger train tardiness and freight train delay are considered in the objective. We model the problem for a general network, where overtaking is allowed whenever the solution can be improved. Besides at stations, the overtaking between trains can also be made at junctions. The number of junctions is much more than the number of stations. Thus due to the flexibility of routing, the scheduling problem becomes computationally complex. In the problem we are studying, the passenger train timetable at intermediate stations is given as a soft constraint, and the schedule is penalized if the actual departure of the passenger train from the station is tardy.

In the literature of train scheduling, while there exists a variety of models, the objective function of the train scheduling problem mainly falls into two subgroups. The models in one subgroup focus on minimizing the travel time or the delay (a common objective when scheduling freight trains). The models in another subgroup focus on minimizing the deviation of scheduled arrival time and actual arrival time (a common objective when scheduling passenger trains). However, to the best of our knowledge, there has been little work that combines the scheduling for passenger and freight trains to jointly consider the travel times of the freight trains and also the

tardiness of the passenger trains, while accounting for the complexity and scalability of real-world railway operation. To address this gap, we present a solution approach involving a joint routing and scheduling model that can be used to minimize the travel times of the freight trains and the tardiness of the passenger trains.

3. Problem Statement and Formulation

The objective is to jointly solve for the passenger and freight rail scheduling problem 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 at which 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. Priority denotes the passing order of the trains over the track segment.

In the studied problem, the passenger train timetable at station stops (intermediate stations) is given, rather than the timetable at each track segment or signal control point. The station timetable is used as a reference to decide the actual arrival and departure times at each track segment or signal control point, as well as the routing and priority decisions of both passenger and freight trains. That is, only the scheduled timestamps at the passenger stations are given. The model then determines the timestamps at various track segments or signal control points to meet the scheduled timestamps at the passenger train stations. This schedule plan is made on a daily basis, or when the freight train demand is changed.

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 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.

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. To describe the problem using a mathematical model, we translate the actual rail network 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. 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. 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. In this paper, we set the length of a node as the longest length of a train and set it to unit capacity. This guarantees at minimum the length of the longest train as the minimum safety headway. Also the position of the junctions, speed limit and the location of the signals are considered in the division of the track segments. The junctions and signals are usually the division point unless the distance between two division points is longer than the longest length of a train. By the definition of unit capacity, each segment and each junction can be occupied by at most one train at any time. As an illustration of the track segment characteristics, Figure 1 and Figure 2 show the railway network and the corresponding arc based network.

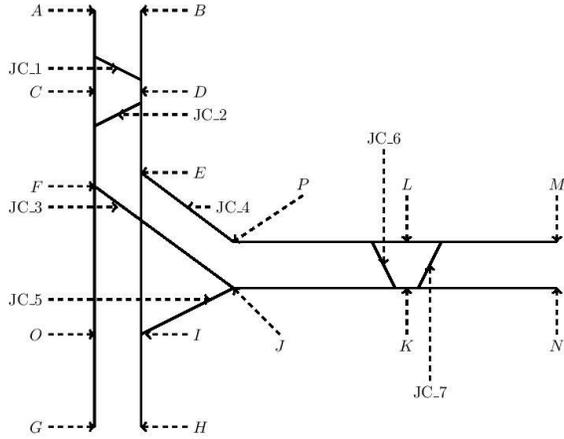


Figure 1. Sample Railway Network

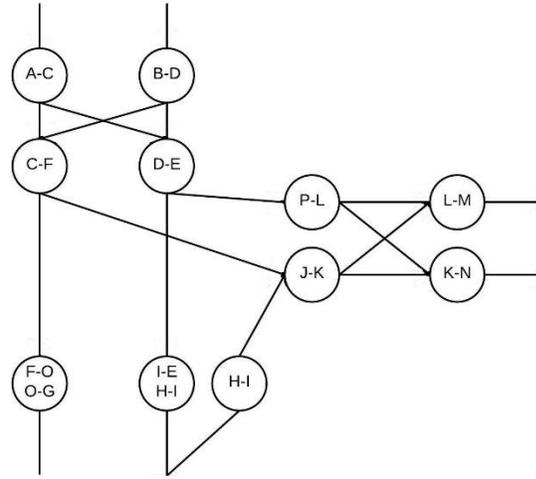


Figure 2. Abstract Graph of Sample Network

The basic components of the network are nodes and arcs. Each node defines a combination of one or more contiguous segments. The node is the basic component of a complete route. Arcs are connections between the nodes, and they represent the movement of trains between nodes. 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.

Corresponding to the model structure, the train information consists of origin and destination, speed limit, length and timetable if it is a passenger train. In order to effectively model and solve this train routing and scheduling problem, we setup metrics to evaluate the passenger trains' tardiness and the freight trains' travel time. We model it as a mixed integer problem, with a set of constraints that guarantee traffic flow conservation, travel time feasibility and a safety headway between the trains. 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 route
D_q	The destination node of train q 's route
S_q	The set of station stop nodes along passenger train q 's route, including origin and destination
N_q^{ED}	The auxiliary dummy end node after destination D_q
N_q^t	The subset of nodes which are reachable for train q
$N_{i,q}^{pre}$	The preceding set of nodes for train q before entering node i
$N_{i,q}^{suc}$	The succeeding set of nodes for train q after exiting node i
$T_{q,s}$	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.

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 at 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 $t_{q,i}^d$.

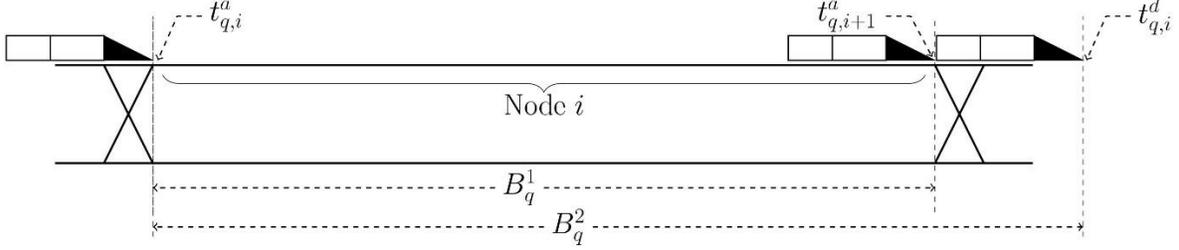


Figure 3. Arrival and Departure Time in Train Movement

Figure 3 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 traffic congestion. To evaluate the time that a train spends on a node, we define two measurements.

$$\begin{aligned} B_{q,i}^1 &= t_{q,i+1}^a - t_{q,i}^a \\ B_{q,i}^2 &= t_{q,i}^d - t_{q,i}^a \end{aligned} \quad (1)$$

$B_{q,i}^1$ denotes the running time and $B_{q,i}^2$ denotes the occupation time, respectively. 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. In our model, $B_{q,i}^1$ and $B_{q,i}^2$ are calculated based on the track segment speed limit, track length, train length and train speed limit. Note that in the calculation of $B_{q,i}^2$, after the train's head leaves the end of node i , the speed limit of node i still applies since part of the train is still in node i . $B_{q,i}^2$ also depends on the succeeding node since speed limit is applied after the train head enters the succeeding node, so we further 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. Our optimization model is presented as follows.

Objective function:

$$\min \sum_{q \in Q_f} (t_{q,D_q}^a - t_{q,O_q}^a) + \sum_{q \in Q_p} \sum_{s \in S_q} \max(t_{q,s}^a - T_{q,s}, 0)$$

Subject to:

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

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

$$\sum_{j \in N_{s, q}^{suc}} I_{q, s, j} = 1, \forall q \in Q_p, s \in S_q \setminus \{O_q, D_q\} \quad (4)$$

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

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

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

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

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

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

$$\left(2 - \sum_{j \in N_{i, q_1}^{suc}} I_{q_1, i, j} - \sum_{k \in N_{i, q_2}^{suc}} I_{q_2, i, k} \right) M + x_{q_1, q_2, i} M + t_{q_1, i}^a \geq t_{q_2, i}^d + \mu, \\ \forall q_1, q_2 \in Q, q_1 \neq q_2, i \in N_{q_1}^t \cap N_{q_2}^t \quad (11)$$

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

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

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

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

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

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

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

The objective function minimizes both the total travel times for the freight trains and the total tardiness for the passenger trains. 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 cannot 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 and the 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 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 types of 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 4, the overall logic of the heuristic algorithm is presented.

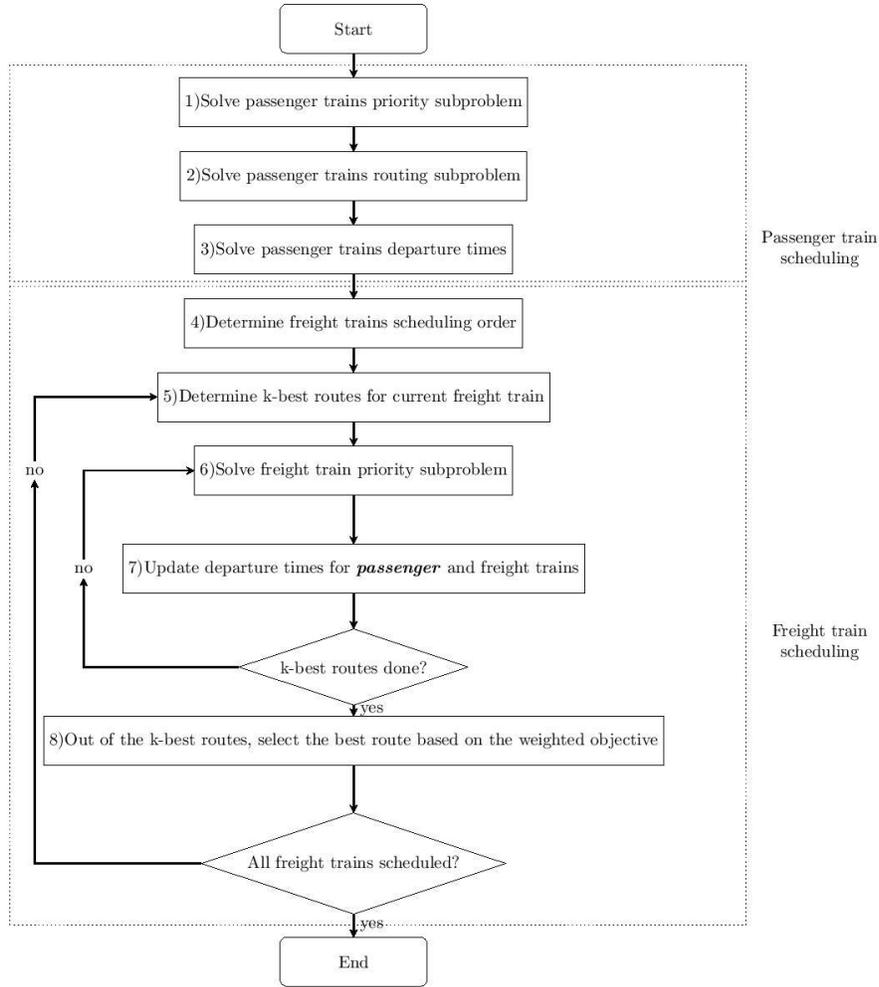


Figure 4. The Flow Chart of the Solution Procedure

The algorithm we propose is based on a two-step decomposition of the original problem. The motivation of decomposing the overall problem is that the size of the MILP is computationally difficult to solve as a single optimization problem. Through a decomposition procedure, several subproblems are solved to produce a heuristic solution to the overall problem. We determine the priority and routing decisions for the passenger trains in the first phase because the station timetables for the passenger trains are given, which can be used as a reference to derive passenger train timetables at non-station nodes (track segments). Furthermore, passenger trains are more sensitive to tardiness. The design of this two phase decomposition is based on the fact that the passenger trains demand is relatively constant in each day; however the freight train demand changes from day to day. Before the daily operation, only the freight train scheduling phase needs to be updated since the passenger train scheduling phase is relatively stable. Once all the routing and priority decisions have been made, the time decisions are made jointly for the passenger and

freight trains. 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 details of the passenger train scheduling subproblem are presented in Section 4.1. Second, freight trains are scheduled sequentially according to a predefined processing order, and this 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.

We note that after the passenger train scheduling phase, the freight trains are scheduled iteratively. At each iteration the priority and routing decisions are determined for the newly scheduled freight train. Note that at this step the freight train could be inserted before or after a passenger train, possibly impacting the arrival times of the passenger trains to their stations. After the priority and routing decisions for this freight train is determined, all the priority and routing decisions for the passenger and the scheduled freight trains to this point are fed into a linear program to determine the time decisions for all these trains. This process is repeated until all freight trains have been scheduled. Thus, after the freight trains are scheduled the arrival times of the passenger trains to the stations may change. Before the scheduling of freight trains, the passenger train solution will be the same on each day. However, different number of freight trains, especially from different origin and destinations, could change the arrival/departure times of the passenger trains at each track segment or signal control point. If the insertion of a freight train fails due to infeasibility, the freight train will be skipped and will not be inserted to the schedule. 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. Note that we only look at the earliness objective when creating the schedule with just passenger trains. The reason we study the earliness of passenger trains is that the earliness at intermediate stations (station stops) indicates the slack time of the passenger trains' arrival compared with the timetable. Optimizing the subproblem based on earliness can allocate the slack time (buffer time) to passenger trains before each intermediate station. Moreover, maximizing the minimal earliness of each passenger train at each station gives a relatively uniform earliness and slack time, thus avoiding an unbalanced schedule in terms of on-time performance. These slack times can then be used when the freight trains are added to the schedule. Note that when solving for the problem when freight trains are added, the objective now considers both passenger train tardiness (i.e., no longer focusing on minimizing earliness since the slack has been previously added) and freight train delay. For the studied rail network, there is sufficient capacity so that the passenger trains can arrive early to their stations without any freight trains, but without careful scheduling this is no longer automatically the case when considering the freight trains. In summary, in the passenger train scheduling phase our approach pushes the passenger train arrival times as early as possible in a balanced way to accommodate for the additional freight trains that need to be scheduled. After adding the freight trains to the schedule, the passenger train arrival times at the stations get pushed forward to closer match the given scheduled timestamps at the stations.

The passenger routing and priority variables (precedence relationship between the passenger trains) that gives the maximum earliness will only need to be solved once as long as the

passenger train demand is unchanged. However, the passenger train time decisions and the remaining priority variables (precedence relationship between the freight and passenger trains) need to be solved daily since the freight train demand changes daily. This is one of the reasons behind the decomposition procedure design.

Subproblem-1:

Objective function:

$$\max_{q \in Q_p, s \in S_q} \min T_{q,s} - t_{q,s}^a$$

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 propose a heuristic rule to solve for the priority decision variables. Since the routes of the passenger trains are flexible, the heuristic rule for priority assignment should be general such that they can fit any of the routes for each passenger train.

First the arrival time of the train at the nodes along its route is estimated. 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 $t_{q,i,r}^a$ 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 $N_{q,i,r}^c$.

Objective function:

$$\min T_{q,i}^a$$

Subject to:

$$t_{q,0,q,r}^a \geq T_{q,0,q}, \forall r \in R_q \tag{19}$$

$$t_{q,j+1,r}^a \geq t_{q,j,r}^a + B_{q,j}^1, \forall j \in N_{q,i,r}^c, r \in R_q \tag{20}$$

$$T_{q,i}^a \geq t_{q,i,r}^a, \forall r \in R_q \tag{21}$$

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 $T_{q,i}^a$ under an arbitrary route, with the assumption of an on-time departure and zero congestion (free flow travel time). However, this estimation of arrival time 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.

The minimal anticipated tardiness can be solved through a similar model, with only a slight transformation. We denote the minimal anticipated tardiness of train q at node i as $T_{q,i}^t$, where $T_{q,i}^t = T_{q,i}^a + T_{q,i,s}^b - T_{q,s}$. $T_{q,i,s}^b$ 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.

Our priority assignment rule is based on the weighted average of the arrival time and the anticipated tardiness. We assign priority mainly according to the earliest arrival time, 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 $T_{q,i}^a$ and anticipated tardiness $T_{q,i}^t$, as a weighted sum time $T_{q,i}^w$:

$$T_{q,i}^w = \alpha * T_{q,i}^a - (1 - \alpha) * T_{q,i}^t, \quad \forall q \in Q_p \quad (22)$$

This weighted measure is used as a heuristic rule to sort the passenger trains based on increasing value of this measure (e.g., smallest has highest priority) on each node to make priority decisions. The weighted coefficients are tuned in the following experiments. The performance of the priority assignment rule depends on the network structure and the density of the passenger train schedules. Experiments should be done to evaluate the appropriate value for α on a specific network and schedule. We use part of the railway network in 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 paper 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 node i . However, the number of routes grows exponentially with the number of junctions. In this sample network, **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 identify the value of α , 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 $T_{q,i}^w$ measure, 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 and maximum of the 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.

Coefficient α		1.0	0.99	0.95	0.9	0.85	0.8	0.7	0.6	0.0
Max-min	Avg	-16.6	-14.3	-15.8	-13.0	-13.0	-16.2	-24.0	-21.7	-120.8
Earliness (min)	Max	0.213	1.098	3.426	3.139	1.825	0.010	-1.415	-0.470	-50.674

Table 1. Performance of Priority Assignment with Different α

From the results, we can conclude that a coefficient setting of $\alpha = 0.9$ is best for this rail network. The primary reason that all of the columns 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 row in Table 1 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 minimize 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 route 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 $|Q_p|$. 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 four criteria in the candidate route set construction:

1. Depending on the travel direction of a train, the track segments are assigned different weights. We assume the left-hand side track along the travel direction is preferred if multiple succeeding nodes are available and are reachable. The weight on the preferred succeeding nodes are assigned twice the weight on the other nodes.
2. Trains are not allowed to make frequent track changes during their travel. Each passenger train is only allowed to do one track change every five times its length.
3. After moving to a non-preferred track, the train attempts to move to a preferred track (left most track segment) within three possible junctions. After three junctions, the train is forced to move back to the preferred track at the earliest feasible time point, and Criteria 1 and 2 will be overridden in this case.
4. The station stops (intermediate stations) have to be passed along the route.

The candidate route set is first filled according to the criteria above. Then based on the total weights along the route, n^c distinct routes with the highest weights are selected as the final candidate set.

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 consist of *selection*, *crossover* and *mutation*, which are defined as follows. The selection method is chosen as tournament selection, since preliminary experiments show that the tournament method performs best to find the individual with the highest fitness value. 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. 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.

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

We use the same sample network as in Section 4.1.2 to perform experiments in selecting the best combination of the 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 2 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 2 use the genetic algorithm for routing with $\alpha = 0.9$. According to this result, $\theta_c = 0.5$ and $\theta_m = 0.2$ are the best settings for these parameters. The result of 5.41 for the Max-Min Earliness compares favorably with the average and maximum results in Table 1. 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. The purpose of presenting Tables 1 and 2 is to illustrate how some of the parameters of the solution algorithm can be calibrated. The intent is not to show that our solution procedure can outperform a random solution procedure. We note however that the best calibrated solution in Table 2 outperforms the best value of the 1,000 random samples of route combinations in Table 1. Since this is a rather large random sample, it is possible to sample one good solution.

Max-Min Earliness				
$\theta_c \backslash \theta_m$	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 2. 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. The priority assignment rules for passenger trains are determined by the anticipated arrival time and tardiness. Freight trains have no station timetable, thus the weighted measure in Equation (22) is no longer applicable. The decision

about freight train priority assignment should minimize the freight train traveling time while having minimal impact on passenger train tardiness.

2. The route requirements are different. The route construction for freight trains should be based on adding minimal traffic congestion to the rail corridor/network. The route construction method developed for passenger trains focused only on maximizing the minimum earliness (i.e., to have as much slack as possible to accommodate the freight trains when they are scheduled) and not on travel time minimization.

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 minimized 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. Here the scheduling order refers to the sequence of handling the freight trains, since the freight trains are inserted sequentially. Train priority refers to the precedence relationship of two trains on a track segment, which corresponds to the decision variables $x_{q_1, q_2, i}$ in Section 3. Given the priority variables of a set of trains, the sequence that trains travel through the node is uniquely determined. The scheduling order does not refer to the ordering of trains visiting a node. It is the order that the freight trains are processed in the solution procedure. In fact, it is possible for a train that is ordered (processed) after another train to be given a higher train priority (precedence relationship) than the earlier ordered (processed) train. 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 first $\{q_1, q_2\}$ from D_1 , $\{q_5\}$ from D_2 , $\{q_7\}$ from D_3 , $\{q_3\}$ from D_1 , $\{q_6\}$ from D_2 and finally $\{q_4\}$ from D_1 .

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 best arrival/departure times.

4.2.2 Freight Train Routing Subproblem

In order for a freight train to be scheduled, its route is first identified. In a general railway network, there are double track, triple track or even more complex track structures. In our network construction, no assumption about track structure is made. The routing of the trains is flexible under the constraints of the moving direction and junction direction.

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 best route could be different at different times of the day. For example, during the rush hour of the day, the best 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:

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

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.

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

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.

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

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. 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 $q^* \in Q_f \setminus Q_f^\Phi$, the search space for its route is within the sub-network that only contains the reduced node set N_q^t . We build another network with the same structure of the sub-network, but the weight of each node is assigned as $F_i + B_{q^*,i}^1$, in which $B_{q^*,i}^1$ 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 newly scheduled freight train to the nodes along a given route. 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. In the freight train scheduling phase, we hold the priority variables between the passenger trains fixed, and insert the freight train into the plan. This freight train can be inserted before or after any previously scheduled freight or passenger train. Note that handling freight trains after passenger trains does not necessarily mean that we always hold back a freight train and let the passenger trains move first. Thus, freight trains can either move before or after passenger trains depending on the situation.

We define the insertion position for train q^* as the position in the sequence of the previously scheduled trains on node i . Figure 5 shows an example of candidate insertion positions on a node. Together, all the given priority variables define a unique sequence of trains. 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.

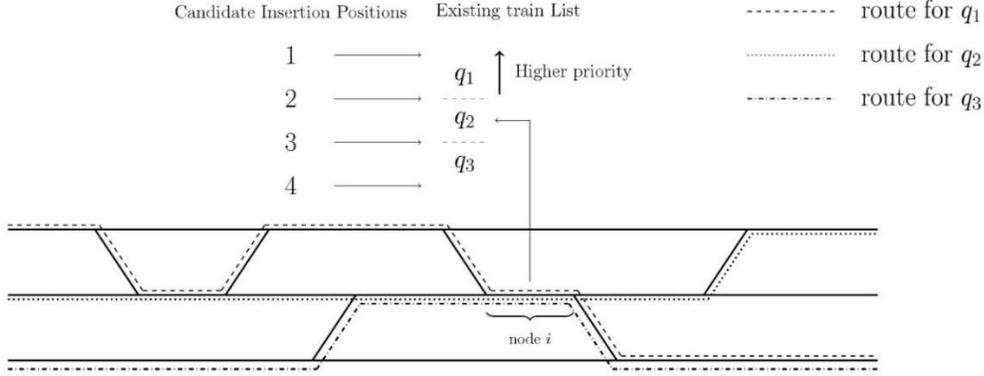


Figure 5. Example of the Candidate Insertion Position Along a Triple Track Segment

We denote the newly scheduled train as q^* and the sequence of nodes along its route as $Z_{q^*}^r$, 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 $Z_{q^*}^r$ 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:

$$\min \sum_{q \in Q_f} (t_{q,D_q}^a - t_{q,O_q}^a) + \sum_{q \in Q_p} \sum_{s \in S_q} \max(t_{q,s}^a - T_{q,s}, 0)$$

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 $j \in Z_{q^*} \setminus \{i\}$. 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 $j \in Z_{q^*} \setminus \{i\}$, we apply a Backward-Forward Insertion (BFI) algorithm to derive their priorities.

4.2.3.1 Freight Train Insertion Rule

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 5 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) $\tilde{t}_{q^*,q',i}^d + \sum_{k=i+1}^j B_{q^*,k}^1 + \mu > t_{q',j}^a, \forall i, j \in Z_{q^*}, i < j - 1, \forall q' \in \hat{S}_{i,q^*} \cap \check{S}_{j,q^*}$
 $\tilde{t}_{q^*,q',i}^d$ is the earliest departure time of q^* from node i after train q' , $\tilde{t}_{q^*,q',i}^d = t_{q',i}^d + \mu + B_{q^*,i,i+1}^2$ and $t_{q',i}^d$ come from the arrival and departure times recorded in the previous iteration.

The proof of **Proposition 1** is presented in the appendix. It 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 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, which is the earliest feasible insertion 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 $Z_{q^*} \setminus \{i\}$. We use the conclusion from **Proposition 1** to eliminate the illegal insertion positions, and the rule of **FIR** is to select the front most 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 $Z_{q^*} \setminus Z_{q^*}^\Phi$ 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^* .

4.2.3.2 Backward-Forward Insertion Algorithm

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 $|Z_{q^*}| - 1$ 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 $a_j^{q^*}, j \in Z_{q^*} \setminus Z_{q^*}^\Phi$ 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 the 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.

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

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

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

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

The above calculations are directly derived from Constraints (6)-(11). 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. When a passenger train is early at its next station and the arrival time at the current node is relaxed to insert freight train q^* , the retiming of the passenger train occurs and the time decisions along the corresponding passenger train's route until the next station are updated by the amount of the relaxed time. In Figure 6, we give an example about the relaxation of the time window based on the early passenger train. 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**.

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

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

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

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

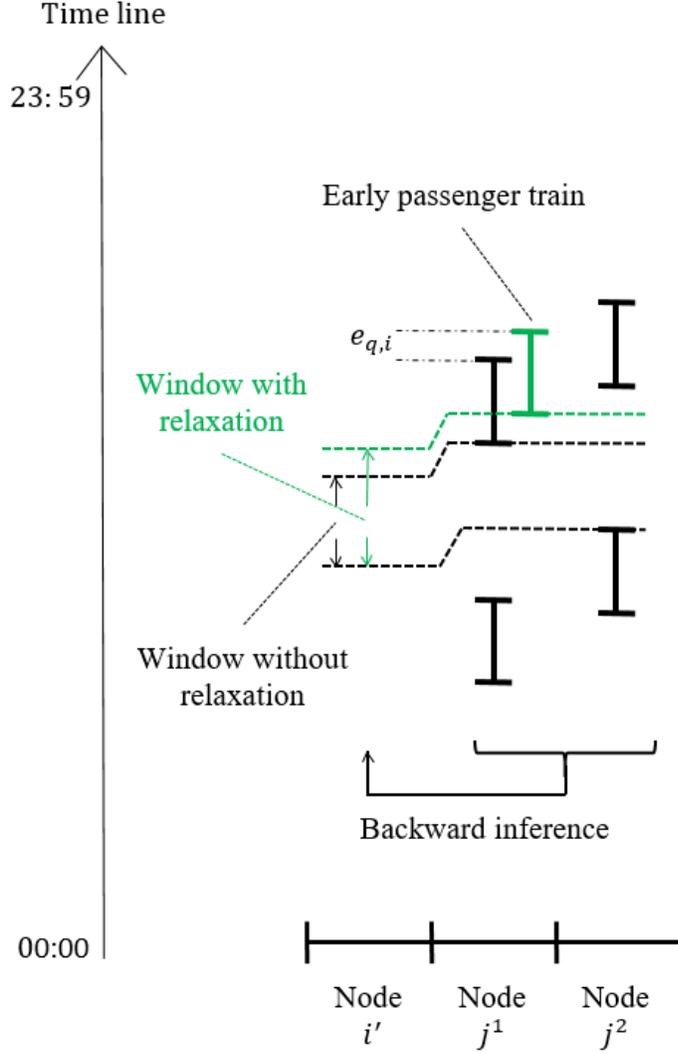


Figure 6. Relaxation of the time window based on an early passenger train

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 $t_{q',i'}^a$ to node i' and departure time $t_{q',i'}^d$ from node i'

If $w_{q^*,i'}^{maxDep} + \mu < t_{q',i'}^a$:

Schedule q^* before q' , $a_{i'}^{q^*} = a_{i'}^{q'}$. Update $Z_{q^*}^\Phi = Z_{q^*}^\Phi \cup \{i'\}$, and go to **Step_1**.

Else if $t_{q',i'}^d + \mu > w_{q^*,i'}^{maxArr}$ and $w_{q^*,i'}^{minDep} + \mu < t_{q',i'}^a$:

Schedule q^* before q' , $a_{i'}^{q^*} = a_{i'}^{q'}$, update $Z_{q^*}^\Phi = Z_{q^*}^\Phi \cup \{i'\}$, and go to **Step_1**.

Else if $w_{q^*,i'}^{minArr} > t_{q',i'}^d + \mu$:

Evaluate next position in $S_{i'}$ and return to **Step_2**.

Else if $w_{q^*,i'}^{minDep} + \mu > t_{q',i'}^a$ and $t_{q',i'}^d + \mu < w_{q^*,i'}^{maxArr}$:

Evaluate next position in $S_{i'}$ and return to **Step_2**

Else if $w_{q^*,i'}^{minDep} + \mu < t_{q',i'}^a$ and $w_{q^*,i'}^{maxArr} > t_{q',i'}^d + \mu$:

$$\text{If } \frac{t_{q',i'}^a - (w_{q^*,i'}^{minDep} + \mu)}{w_{q^*,i'}^{maxDep} - w_{q^*,i'}^{minDep}} > \frac{w_{q^*,i'}^{maxArr} - (t_{q',i'}^d + \mu)}{w_{q^*,i'}^{maxArr} - w_{q^*,i'}^{minArr}}.$$

Schedule q^* before q' , $a_{i'}^{q^*} = a_{i'}^{q'}$, update $Z_{q^*}^\Phi = Z_{q^*}^\Phi \cup \{i'\}$, and go to **Step_1**.

Else: Evaluate next position in $S_{i'}$ and return to **Step_2**.

Step_4: Return the insertion positions $\{a_1^{q^*}, a_2^{q^*} \dots a_{i'}^{q^*}\}$

In Figure 7, 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. The axis shows the window of arrival time and departure time. Each row corresponds to a case in **Step_3**, in which the segment stands for the time interval of arrival and departure of train q' , with an extended segment to the right which stands for the safety headway μ .

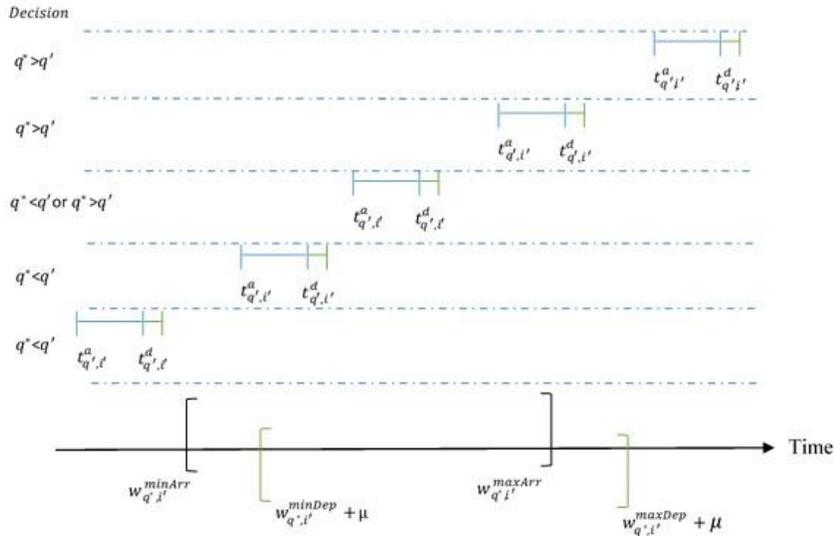


Figure 7. Inference Rules in **Step_3**

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 $w_{q^*,i'}^{minDep}(j)$ and $w_{q^*,i'}^{maxDep}(j)$ remains the same as the backward inference step, the calculation for $w_{q^*,i'}^{minArr}(j)$ and $w_{q^*,i'}^{maxArr}(j)$ changes to

$$w_{q^*,i'}^{minArr}(j) = \begin{cases} 0 & a_j^{q^*} = 1, j \in Z_{q^*}^\Phi \\ \min(t_{\check{q},j}^d + \mu + \sum_{l=j+1}^{i'} B_{q^*,l}^1, T^E) & a_j^{q^*} > 1, j \in Z_{q^*}^\Phi \end{cases}$$

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

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 in $\{Q_p \cup Q_f^\Phi\}$. 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 obtained from the solution.

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 $j \in Z_{q^*} \setminus \{i\}$ 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 $j \in Z_{q^*} \setminus \{i\}$. 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 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 4.1. We add 84 freight trains and 22 demand sets to the previous data sets. For the k-best routes, we set $k = 1$ to compare the performances on one route. 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.

4.2.3.3 BFI Algorithm Experiments

Table 3 shows the comparison of the objective function and CPU time between the two algorithms. Each row is the result of scheduling one additional freight train, and each row depends on the priority sequences of the previously scheduled trains. 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.

	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	81.0	0.7	81.0	0.3	30	2078.8	18.1	2091.4	2.7	59	3874.8	31.2	3880.6	15.9

2	159.4	0.8	159.4	0.4	31	2143.8	7.5	2166.6	3.4	60	3950.2	32.3	3942.8	13.0
3	233.8	0.9	233.8	0.4	32	2214.8	8.4	2229.8	3.6	61	4010.0	33.1	4005.0	13.5
4	301.4	1.1	301.4	0.4	33	2279.8	8.6	2292.2	3.3	62	4077.4	35.9	4067.2	17.7
5	371.0	1.0	371.0	0.5	34	2344.0	8.7	2356.6	4.4	63	4142.2	35.2	4133.0	12.7
6	435.0	1.1	436.6	0.5	35	2405.0	10.3	2419.8	6.7	64	4217.4	40.2	4196.0	16.5
7	507.0	1.3	507.0	0.6	36	2471.6	10.1	2480.8	5.8	65	4262.4	41.2	4239.4	10.8
8	582.8	1.4	582.8	0.7	37	2532.2	14.1	2546.6	2.7	66	4303.2	39.3	4284.4	9.2
9	653.4	4.6	664.6	0.7	38	2593.6	6.2	2606.6	10.3	67	4385.2	38.5	4350.0	19.5
10	730.4	2.3	730.6	0.8	39	2656.0	12.4	2671.2	7.4	68	4450.0	45.3	4413.8	17.6
11	795.4	3.9	796.2	0.9	40	2731.6	9.7	2745.4	6.8	69	4521.0	46.3	4491.2	17.5
12	860.6	7.8	861.6	0.9	41	2796.8	13.8	2809.2	4.7	70	4584.0	47.0	4555.4	14.2
13	931.0	13.1	932.0	1.2	42	2857.8	13.8	2876.0	5.4	71	4659.4	48.6	4618.2	14.0
14	991.8	7.4	995.2	1.0	43	2901.8	9.3	2939.0	6.3	72	4722.6	49.3	4694.0	13.5
15	1063	3.0	1066.4	1.5	44	2945.2	10.8	3003.4	6.7	73	4790.4	50.8	4760.6	13.2
16	1134.4	3.6	1138.2	1.5	45	3003.0	7.7	3065.8	4.6	74	4866.2	51.3	4828.4	18.2
17	1201.2	4.1	1204.6	1.7	46	3046.6	10.2	3120.2	4.1	75	4935.4	54.3	4895.2	18.1
18	1270.0	3.0	1274.2	1.7	47	3086.2	10.5	3163.2	5.2	76	4999.4	52.4	4959.2	18.1
19	1336.2	9.2	1354.2	1.9	48	3125.8	15.7	3214.8	6.4	77	5065.4	51.3	5020.8	17.3
20	1413.4	5.5	1415.8	2.6	49	3204.8	8.6	3250.2	4.8	78	5134.8	50.3	5082.8	18.0
21	1479.8	5.4	1482.2	3.1	50	3271.0	10.4	3291.0	5.6	79	5204.0	53.5	5150.4	17.4
22	1546.0	6.8	1548.4	3.6	51	3339.0	6.4	3335.2	7.2	80	5267.0	54.9	5215.6	17.0
23	1612.4	15.1	1616.2	3.3	52	3407.2	10.7	3416.8	7.2	81	5337.2	55.5	5283.2	21.3
24	1680.4	12.5	1682.8	3.5	53	3473.8	19.3	3481.0	13.6	82	5411.0	56.8	5348.4	13.8
25	1749.2	11.1	1751.2	3.3	54	3541.8	20.1	3545.4	16.6	83	5472.0	58.5	5410.8	15.1
26	1814.6	13.0	1817.6	4.5	55	3606.8	20.0	3612.4	11.8	84	5533.0	58.8	5472.8	23.1
27	1880.2	10.5	1882.8	2.2	56	3675.8	17.0	3681.4	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 3. Comparison Between BFI and Opt-Integer with Same Passenger Schedule

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 solutions 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.

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 58 nodes and 63 arcs. There are a total of 5 passenger trains. The railway trackage is show in Figure 8 and the timetable for the two schedules are shown in Table 4. 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. The numbers in Figure 8 refer to the length (in miles) of the track segments.

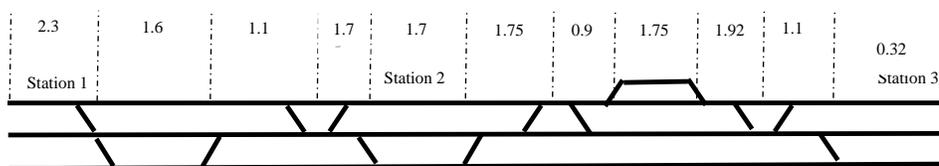


Figure 8. Test Network with Three Stations

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 4. Test Passenger Train Schedule

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. We set $k = 2$ in this section. The results are presented in Table 5. 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 the number of freight trains 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. To illustrate how the solution approach distributes the use of the different track segments in Figure 8, we plot the number of passing trains per node in Figure 9 in the Appendix for the Uniform-Schedule scenario.

# of Freight Trains	# of integer variables	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	1,615	98.234	0.0	40.1	98.444	82.1	98.208	0.0	57.1	101.036	90.6
5	2,063	120.312	0.0	413.8	128.473	95.7	118.686	0.0	695.1	121.437	92.8
6	2,538	147.314	0.0	7,508	149.061	96.4	147.314	0.0	7,692	147.656	97.6
7	3,102	167.9	2,847	10,000	168.239	100.2	169.52	3510	10,000	169.538	98.1
8	3,677	198.27	3,531	10,000	198.272	105.3	196.634	4240	10,000	202.118	110.9

Table 5. Comparison Between Optimal-solution and Heuristic-Solution

5.2 Large network

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. We present the origin/destination (OD) pairs and the layout of the network in the Appendix. Several trains types are assigned. The average length of a passenger train is 750 feet and the average length of a freight train is 6,000 feet.

To schedule the train 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 First-Come-First-Serve 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*. During the deceleration process, the availability of the successor node is checked and the status of the train is updated dynamically. In another approach, first the passenger train schedules are solved using our proposed method in Section 4.1, then the freight trains are sequentially inserted into the schedule assuming the schedules of the previously scheduled passenger trains are fixed. That is, all time decisions for the passenger trains at all the nodes (e.g., stations, junctions, and signal points) are fixed and are treated as hard constraints when scheduling the freight trains. This approach is referred to as *Sequential* since the passenger train schedule is determined first and held completely fixed and then the freight train schedules are determined. The fourth approach is to assign the freight trains with uniform departure times (equal interval) and use the greedy algorithm for routing and priority assignment, which is referred as *Uniform-Departure*. In the *Uniform-Departure* approach, the departure times of passenger trains from the origin station follow the timetable.

A comparison of the four approaches is presented in Table 6. Freight train delay is the difference between the actual travel time and the free flow travel time. In Table 6, we present two average delays for the freight: one when there are no passenger trains in the network (average delay 1) and the other where there are passenger trains in the network (average delay 2). The difference between these two average delays shows the impact on the freight trains on average when the passenger trains share the same track infrastructure. 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. In this set of experiments, we randomly generate the origin-destination combination for each freight train. For each given number of freight trains, we run five random samples and report the average statistics in Table 6.

In Table 6, the *Complete-Control* control approach outperforms the other three approaches in reducing the freight train average delay and the passenger train tardiness. In the *Uniform-Departure* 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. Compared with the *Complete-Control*, the *Sequential* has larger freight train average delay. Comparing Average Delay 1 with Average Delay 2, the results show the importance of jointly scheduling passenger and freight trains since in the *Sequential* and *Uniform-Departure* heuristics there is a significant difference between these two average delays. That is, in the presence of passenger trains, the average delay for freight trains is significantly larger for these heuristics. For the proposed *Complete-Control* approach, the average freight train delay does not increase that significantly when passenger trains are introduced to the rail network. The passenger train tardiness is zero for the *Sequential method* since without considering the freight trains there is sufficient capacity in the system to ensure on-time arrivals, but since the passenger time decisions are treated as fixed after the passenger scheduling phase, the freight train scheduling problem is more constrained and the arrival/departure times of all the trains are not jointly updated as in *Complete-Control*; thus the encountered traffic congestion generates a higher delay. 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 demand, our proposed solution procedure provides an efficient and high quality solution for the joint passenger and freight train scheduling and routing problem.

# of Freight Train	Complete-Control				Partial-Control			
	Freight train		Passenger train		Freight train		Passenger train	
	Average Delay 1 (min)	Average Delay 2 (min)	Tardiness Percentage (%)	Average Tardiness (min)	Average Delay 1 (min)	Average Delay 2 (min)	Tardiness Percentage (%)	Average Tardiness (min)
84	0.0	1.612	0	0	1.035	3.491	5.03	1.191
95	0.0	1.701	0	0	1.421	3.676	8.11	1.278

106	0.0	1.739	0	0	1.982	3.991	8.32	1.301
117	0.125	1.983	0	0	2.312	5.122	8.95	1.319
128	0.179	2.001	0	0	2.613	5.787	10.17	1.401
139	0.593	2.107	0	0	3.052	7.013	10.55	1.513
150	0.826	2.311	0	0	3.421	9.134	11.24	1.537
# of Freight Train	Sequential				Uniform-Departure			
	Freight train		Passenger train		Freight train		Passenger train	
	Average Delay 1 (min)	Average Delay 2 (min)	Tardiness Percentage (%)	Tardiness Percentage (%)	Average Delay 1 (min)	Average Delay 2 (min)	Tardiness Percentage (%)	Average Tardiness (min)
84	1.077	4.175	0	0	2.4377	9.134	26.11	5.762
95	1.601	4.331	0	0	2.724	10.895	29.39	5.913
106	2.085	5.102	0	0	3.039	11.342	36.21	7.354
117	2.513	6.119	0	0	3.963	12.341	39.00	7.822
128	3.147	8.284	0	0	4.762	12.908	39.98	8.657
139	4.532	9.391	0	0	5.930	13.429	42.89	10.989
150	5.609	10.917	0	0	7.037	15.107	48.56	11.860

Table 6. Performance Comparison Between Complete-Control, Partial-Control, Sequential and Uniform-Departure

6 Conclusions

In this paper, we present a solution approach involving a joint 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 5, 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. The benefits of decomposing the scheduling comes in two aspects. The first is that by decomposing the problem, large size real world problems are solvable in a reasonable amount of computation time. The un-decomposed approach (solving it as one big optimization problem) makes the problem computationally hard to solve for daily operation. Nevertheless, in Table 5, we compare our decomposed procedure with an un-decomposed approach on a small problem size where it is possible to solve the un-decomposed approach. For larger instances, it is not computationally feasible to solve using an un-decomposed

optimization approach. Thus, we compare our decomposed approach with other possible heuristics show that our heuristic outperforms other heuristic rules.

Acknowledgement

We acknowledge Metrans and the Volvo Research and Education Foundation for its kind support of this research.

References

- Arenas, D., Rémy C., Said H., and Joaquin R., 2015. Solving the Train Timetabling Problem, a mathematical model and a genetic algorithm solution approach. *6th International Conference on Railway Operations Modelling and Analysis (RailTokyo2015)*.
- Bersani, C., Qiu, S., Sacile, R., Sallak, M. and Schön, W., 2015. Rapid, robust, distributed evaluation and control of train scheduling on a single line track. *Control Engineering Practice*, 35, pp.12-21
- Bešinović, N., Goverde, R.M., Quaglietta, E. and Roberti, R., 2016. An integrated micro–macro approach to robust railway timetabling. *Transportation Research Part B: Methodological*, 87, pp.14-32.
- Brännlund, U., Lindberg, P.O., Nõu, A. and Nilsson, J.E., 1998. Railway timetabling using Lagrangian relaxation. *Transportation Science*, 32(4), pp.358-369.
- Caprara, A., Fischetti, M. and Toth, P., 2002. Modeling and solving the train timetabling problem. *Operations Research*, 50, 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, pp.738-753.
- Cacchiani, V., Caprara, A. and Toth, P., 2008. A column generation approach to train timetabling on a corridor. *4OR*, 6(2), pp.125-142.
- Cacchiani, V., Caprara, A. and Toth, P., 2010. Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological*, 44, pp.215-231.
- Cacchiani, V., Furini, F., and Kidd, M. P. 2016. Approaches to a real-world train timetabling problem in a railway node. *Omega*, 58, pp. 97-110.
- Caprara, A., Galli, L. and Toth, P., 2007. Solution of the train platforming problem. *In OASICS-OpenAccess Series in Informatics (Vol. 7)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

- Cordeau, J.F., Toth, P. and Vigo, D., 1998. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32, pp.380-404.
- Corman, F., D'Ariano, A., Hansen, I.A. and Pacciarelli, D., 2011. Optimal multi-class rescheduling of railway traffic. *Journal of Rail Transport Planning & Management*, 1(1), pp.14-24.
- 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, pp.141-152.
- Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numerische mathematik*, 1, pp.269-271.
- Eppstein, D., 1998. Finding the k shortest paths. *SIAM Journal on Computing*, 28, pp.652-673.
- Flier, H., Thomas G., and Marc N., 2009. Scheduling additional trains on dense corridors. *International Symposium on Experimental Algorithms. Springer Berlin Heidelberg*.
- Goverde, R.M., Bešinović, N., Binder, A., Cacchiani, V., Quaglietta, E., Roberti, R. and Toth, P., 2016. A three-level framework for performance-based railway timetabling. *Transportation Research Part C: Emerging Technologies*, 67, pp.62-83.
- Hansen, I.A., 2010. State-of-the-art of railway operations research. *Timetable planning and information quality' (WITPress, 2010)*, pp.35-48.
- Harrod, S.S., 2012. A tutorial on fundamental model structures for railway timetable optimization. *Surveys in Operations Research and Management Science*, 17, pp.85-96.
- Karoonsoontawong, A. and Taptana, A., 2015. Branch-and-bound-based local search heuristics for train timetabling on single-track railway Network. *Networks and Spatial Economics*, pp.1-39.
- Lamorgese, L. and Mannino, C., 2015. An exact decomposition approach for the real-time train dispatching problem. *Operations Research*, 63, pp.48-64.
- Lamorgese, L., Mannino, C., and Piacentini, M., 2016. Optimal train dispatching by benders'-like reformulation. *Transportation Science*, 50, pp. 910 - 925.
- Liebchen, C., and Möhring. R.H., 2007. The modeling power of the periodic event scheduling problem: railway timetables—and beyond. *Algorithmic methods for railway optimization. Springer Berlin Heidelberg*, pp. 3-40.
- Liu, S.Q. and Kozan, E., 2011. Scheduling trains with priorities: a no-wait blocking parallel-machine job-shop scheduling model. *Transportation Science*, 45, pp.175-198.

- Louwerse, I. and Huisman, D., 2014. Adjusting a railway timetable in case of partial or complete blockades. *European Journal of Operational Research*, 235, 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, pp.48-75.
- Lusby, R.M., Larsen, J., Ehrgott, M. and Ryan, D., 2011. Railway track allocation: models and methods. *OR Spectrum*, 33, pp.843-883.
- Mascis, A. and Pacciarelli, D., 2002. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143, 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, pp.246-274.
- Meng, L. and Zhou, X., 2011. Robust single-track train dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach. *Transportation Research Part B: Methodological*, 45(7), pp.1080-1102.
- 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.
- Mu, S. and Dessouky, M., 2011. Scheduling freight trains traveling on complex networks. *Transportation Research Part B: Methodological*, 45, pp.1103-1123.
- Oliveira, E. and Smith, B.M., 2000. A job-shop scheduling model for the single-track railway scheduling problem. *Research Report Series-University of Leeds School Of Computer Studies LU SCS RR*, (21)
- Roy, B. and Sussman, B., 1964. Les problèmes d'ordonnancement avec contraintes disjonctives, Note DS no. 9 bis, SEMA, Paris.
- Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J. and Pacciarelli, D., 2016. Ant colony optimization for the real-time train routing selection problem. *Transportation Research Part B: Methodological*, 85, pp.89-108.
- Sama, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., and Pacciarelli, D., 2015. A Routing Filter for the Real-time Railway Traffic Management Problem Based on Ant Colony Optimization. *Transportation Research Procedia*, 10, pp. 534-543.

Semet, Y., and Schoenauer, M., 2005. An efficient memetic, permutation-based evolutionary algorithm for real-world train timetabling. *2005 IEEE Congress on Evolutionary Computation*, 3, pp. 2752-2759

Šemrov, D., Marsetič, R., Žura, M., Todorovski, L. and Srdic, A., 2016. Reinforcement learning approach for train rescheduling on a single-track railway. *Transportation Research Part B: Methodological*, 86, pp.250-267.

Schachtebeck, M. and Schöbel, A., 2010. To wait or not to wait-and who goes first? Delay management with priority decisions. *Transportation Science*, 44(3), pp.307-321.

Shafia, M.A., Aghaee, M.P., Sadjadi, S.J. and Jamili, A., 2012. Robust Train Timetabling problem: Mathematical model and branch and bound algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 13, pp.307-317.

Sun, Y., Cao, C. and Wu, C., 2014. Multi-objective optimization of train routing problem combined with train scheduling on a high-speed railway network. *Transportation Research Part C: Emerging Technologies*, 44, pp.1-20.

Veelenturf, L. P., Kidd, M. P., Cacchiani, V., Kroon, L. G., and Toth, P., 2016. A railway timetable rescheduling approach for handling large-scale disruptions. *Transportation Science*, 50, pp. 841 – 862.

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, pp.320-341.

Appendix

Proof of Proposition 1:

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),

$$\begin{cases} t_{q^*,i}^a \geq t_{q',i}^d + \mu \\ t_{q',i-1}^a \geq t_{q^*,i-1}^d + \mu \end{cases} \quad (26)$$

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

$$\left. \begin{aligned} t_{q,i}^a - t_{q,i-1}^a &\geq B_{q,i}^1 \Rightarrow t_{q',i}^a > t_{q',i-1}^a \\ t_{q',i}^d - t_{q',i-1}^d &\geq B_{q',i}^1 \Rightarrow t_{q',i}^d > t_{q',i-1}^d \\ t_{q',i-1}^d - t_{q',i}^a &\geq B_{q,i-1,i}^2 - B_{q,i}^1 \Rightarrow t_{q',i-1}^d > t_{q',i}^a \end{aligned} \right\} \Rightarrow t_{q',i}^d > t_{q',i-1}^a \quad (27)$$

By Constraints (9),

$$t_{q^*,i-1}^d - t_{q^*,i}^a \geq B_{q^*,i-1,i}^2 - B_{q^*,i}^1 \Rightarrow t_{q^*,i-1}^d > t_{q^*,i}^a \quad (28)$$

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

$$\begin{aligned} t_{q',i}^d &> t_{q',i-1}^a > t_{q^*,i-1}^d \\ t_{q^*,i-1}^d &> t_{q^*,i}^a > t_{q',i}^d \end{aligned}$$

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

$$\left\{ \begin{aligned} t_{q^*,i-1}^d - t_{q^*,i}^a &\geq B_{q^*,i-1,i}^2 - B_{q^*,i}^1 \\ t_{q',i}^d - t_{q',i-1}^a &\geq B_{q',i,i-1}^2 - B_{q',i}^1 \end{aligned} \right\} \Rightarrow \left\{ \begin{aligned} t_{q^*,i-1}^d &> t_{q^*,i}^a \\ t_{q',i}^d &> t_{q',i-1}^a \end{aligned} \right. \quad (29)$$

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

$$t_{q',j}^a \geq t_{q^*,j}^d + \mu \quad (30)$$

For the statement of Condition (c), we have

$$\tilde{t}_{q^*,q',i}^d + \sum_{k=i+1}^j B_{q^*,k}^1 + \mu > t_{q',j}^a \quad (31)$$

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

$$t_{q^*,i}^d > \tilde{t}_{q^*,q',i}^d \quad (32)$$

From (31) and (32), we have

$$t_{q^*,i}^d + \sum_{k=i+1}^j B_{q^*,k}^1 + \mu > t_{q',j}^a \quad (33)$$

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

$$t_{q^*,j}^d \geq t_{q^*,i}^d + \sum_{l=i+1}^j B_{q^*,l}^1 \quad (34)$$

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

$$t_{q^*,j}^d + \mu > t_{q',j}^a$$

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

Number of passing trains per node in the sample network with 58 nodes:

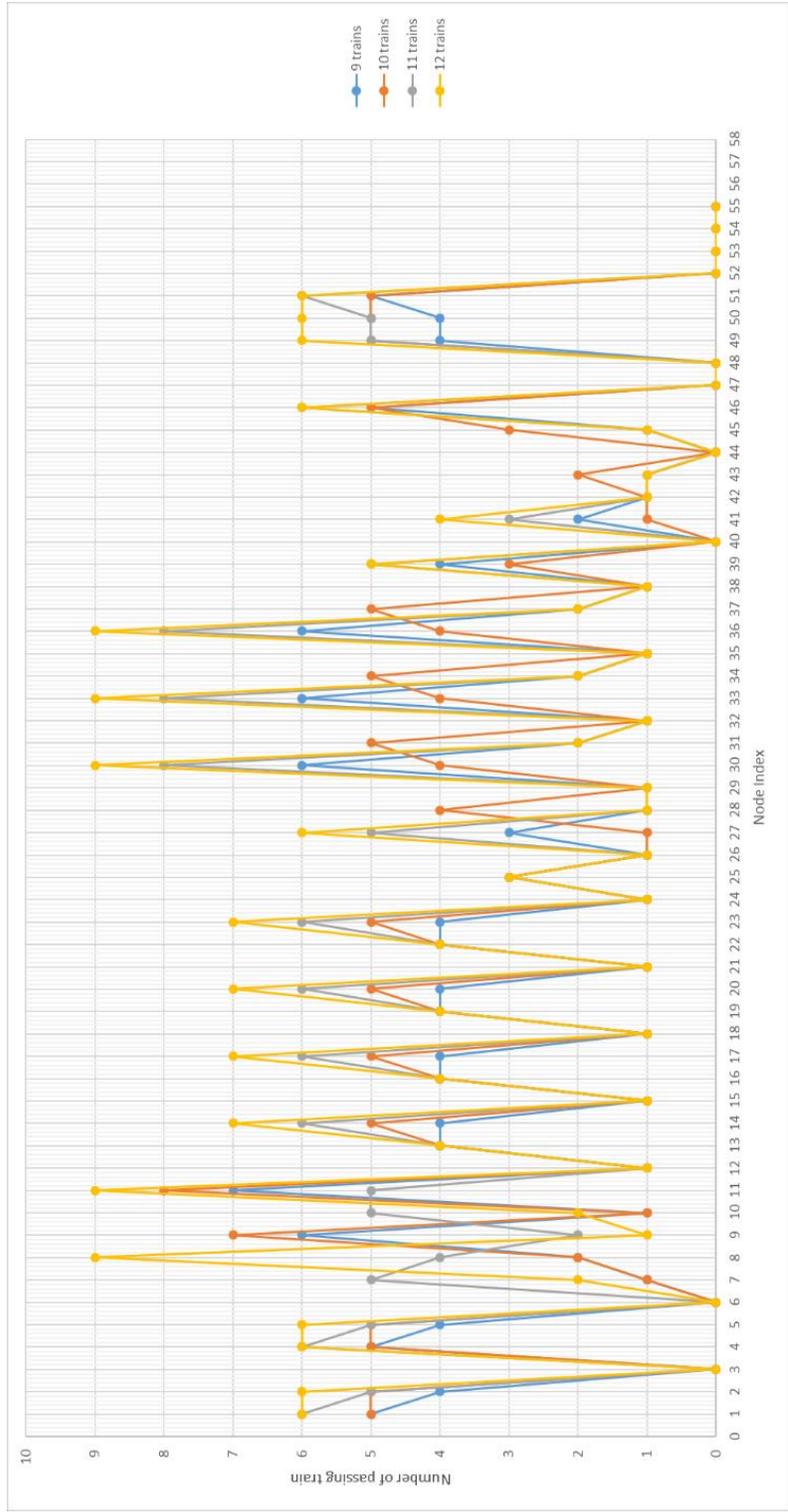


Figure 9. Number of passing trains per node for the Uniform-Schedule

Layout of the large network:

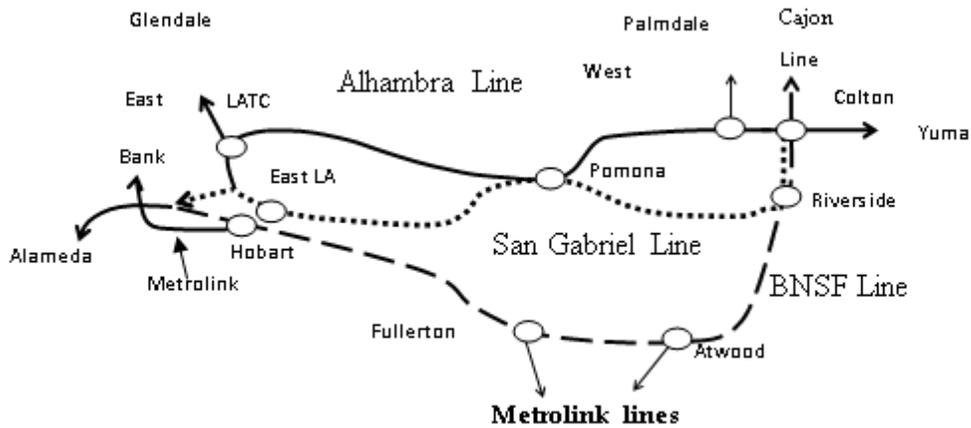


Figure 11. Layout of the large network in Section 5.2

OD pairs for the large test network:

	Origin station	Destination Station	Number of trains per day
Freight train	Cajon	Alameda Corridor	23
	Alameda Corridor	Cajon	23
	Cajon	Hobart Yard	16
	Hobart Yard	Cajon	16
	AtWood	Cajon	3
	Cajon	AtWood	3
	Cajon	Alameda Corridor	23
Passenger train	Union Station	Cajon	5
	Cajon	Union Station	5
	Union Station	Fullerton	25
	Fullerton	Union Station	28
	West Corona	Riverside	4
	Riverside	West Corona	4
	Cajon	West Corona	4
	West Corona	Cajon	4
	Union Station	Riverside	5
	Riverside	Union Station	5

Table 7. OD pairs for the large test network