https://doi.org/10.1016/j.trb.2018.09.015

Integrated Train Timetabling and Locomotive Assignment

Xiaoming Xu*

Chung-Lun Li†

Zhou Xu‡

March 19, 2018

Revised August 20, 2018

Revised September 28, 2018

Abstract

Train timetabling and locomotive assignment are often performed separately in a sequential manner. One obvious disadvantage of such hierarchical planning process is that it often results in poor coordination between the train schedule and the locomotive schedule. This paper focuses on modeling and solving an integrated train timetabling and locomotive assignment problem. To solve this integrated problem, we first construct a three-dimensional state-space-time network in which a state is used to indicate which train a locomotive is serving. We then formulate the problem as a minimum cost multi-commodity network flow problem with incompatible arcs and integer flow restrictions. We present a Lagrangian relaxation heuristic for solving this network flow problem. We conduct a computational study to test the effectiveness of our Lagrangian relaxation heuristic, compare the performance of our heuristic with that of two benchmark solution methods, and report the benefits obtained by integrating train timetabling and locomotive assignment decisions.

Keywords: Train timetabling; locomotive assignment; routing; state-space-time network; Lagrangian relaxation

*School of Automotive and Transportation Engineering, Hefei University of Technology, Hefei, China; email: *xmxu@hfut.edu.cn*

†Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong; email: *chung-lun.li@polyu.edu.hk*. Corresponding author.

‡Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong; email: *zhou.xu@polyu.edu.hk*

1 Introduction

Due to the complexity of the railway planning process, different planning tasks are often performed separately in a sequential manner. Typically, train scheduling decisions, such as train path determination and train timetabling, are made before locomotives and other rolling stocks are assigned to train services. In other words, when train timetabling is performed, the locomotives' schedule and routes are neglected. As mentioned by Bussieck et al. (1997), such hierarchical planning process poses an obvious disadvantage "since the optimal output of a subtask which serves as the input of a subsequent task, will not result, in general, in an overall optimal solution." Thus, separating train timetabling from locomotive assignment decisions often lead to poor coordination between the train schedule and the locomotive schedule. This may result in delay in the locomotives' arrival at the trains' departure stations, which in turn results in delay in the trains' departure, causing the need to revise the entire pre-planned train timetable. This may also result in early arrival of locomotives at the trains' departure stations, leading to low utilization of railway resources.

In this paper, we present an integrated optimization model that simultaneously determines the train timetable, the assignment of locomotives to trains, and the routes for the locomotives when they are light-running (i.e., when they are not serving any train). We assume that each train has a pre-specified traveling route, each locomotive has a given initial location and a given final destination, and that a train can be assigned any locomotive that satisfies the power requirement of the train. We develop a Lagrangian relaxation heuristic for this model and test the efficiency and effectiveness of this heuristic via a computational study.

Train timetabling is an important area of railway operation. A train timetabling problem (TTP) aims to determine the arrival and departure times of each train at each station along the given train path, such that certain operational requirements such as track capacity constraints, minimum headway constraints, etc., are taken into consideration. There has been extensive literature on mathematical models and solution methods for TTPs on both single track networks (see, e.g., Szpigel 1973, Cai and Goh 1994, Caprara et al. 2002, Zhou and Zhong 2007) and general railway networks (see, e.g., Cacchiani et al. 2010a, Cadarso et al. 2015, Zhu et al. 2014). Although different TTP papers typically consider different versions of the problem, the problem is generally referred to as being NP-hard (Caprara et al. 2002). Hence, different heuristic and simulation approaches have been developed for obtaining near-optimal solutions. See Lusby et al. (2011) and Kroon et al. (2014) for comprehensive reviews on TTPs.

Locomotive planning is another important area in railway operation. Following the definition of Vaidyanathan et al. (2008a), the locomotive planning problem (LPP) "considers a fixed pre-defined train schedule and concerns the optimal management of locomotives in order to deliver each train in the schedule from its origin to its destination at minimal cost." Here, the LPP includes one or both of the following two decisions: (i) Locomotive assignment decision; that is, assigning locomotives or locomotive consists (i.e., sets of locomotives) to trains in such a way that no two trains are served by the same locomotive simultaneously. In some models, the assignment of more than one locomotive or one locomotive consist to serve a train is allowed (e.g., using one locomotive to pull the train for a certain distance and using another locomotive to continue pulling that train for another distance). (ii) Locomotive routing decision; that is, determining the sequence of trains that each locomotive serves. This decision also includes the determination of the routes for the light-running locomotives, as locomotives often need to travel from one station to another between services. In some models, additional considerations such as fueling and servicing needs are included. Some models also allow deadheading, where deadheading locomotives do not pull any train but are repositioned by the active locomotives.

A great deal of research on locomotive planning is available in the literature. Some research works consider LPPs with homogeneous locomotives; see, for example, Ghoseiri and Ghannadpour (2010). Some works consider LPPs in which there are multiple types of locomotives but each train is pulled by a single locomotive; see, for example, Booler (1980), Wright (1989), and Forbes et al. (1991). Some works consider LPPs in which more than one locomotive can be assigned to a single train; see, for example, Florian et al. (1976), Ziarati et al. (1997, 1999), Ahuja et al. (2005), and Vaidyanathan et al. (2008a, 2008b). Some works have incorporated car assignment decision into locomotive planning. Cordeau et al. (2000) studied a problem that simultaneously assigns appropriate locomotives and cars to pre-scheduled passenger trains. Cordeau et al. (2001a, 2001b) extended the work of Cordeau et al. (2000) by introducing equipment substitution possibilities, maintenance constraints, and car switching penalties. Other related works include Godwin et al. (2006) who propose a heuristic method for assigning locomotives to freight trains in which locomotive assignment and freight train schedules are developed in two separate phases, Kuo and Nicholls (2007) who study the allocation of locomotives to yards and the moving of light engines between yards, and Nourbakhsh and Ouyang (2010) and Raviv and Kaspi (2012) who study the optimal fueling strategies for locomotives. Comprehensive reviews on LPPs can be found in Cordeau et al. (1998) and Piu and Speranza (2014). Our work differs from traditional locomotive planning research in that our model makes train timetabling and locomotive planning decisions simultaneously, while most locomotive planning studies focus on locomotive assignment and/or locomotive routing decisions with a pre-planned train timetable.

Integrated models for train timetabling and rolling stock scheduling have received increasing attention recently. Dauzère-Pérès et al. (2015) examined the integrated planning of rolling stock and train drivers when a set of train paths is given. Zhu et al. (2014) analyzed a freight rail transportation problem which integrates train service selection decision, train make-up decision (i.e., decision on grouping loaded and empty cars into blocks and grouping blocks into trains), and freight routing decision. Cadarso et al. (2015) considered an integrated model for the recovery of train timetable and rolling stock schedules when the railway system encounters a disruption. Xu et al. (2017) considered an integrated scheduling problem of locomotives and trains on a single track railway line. Our research topic bears some similarities with those of Zhu et al. and Cadarso et al. as we also integrate the planning decisions of train timetabling and rolling stocks. Zhu et al.'s model focuses on train service selection and train make-up decisions, while Cadarso et al.'s model focuses on train service selection and recovery planning. Our model, however, does not include train make-up decisions or recovery planning, but includes the construction of train timetables as well as routing decisions of light-running locomotives, which are features not covered by these two models. Similar to Xu et al.'s model, our model makes train scheduling and locomotive assignment decisions simultaneously, but our model involves complicated routing decisions as we consider a general underlying railway network, while Xu et al. focus on the planning of a single track railway line in which a relatively simple simulation-based heuristic can be applied.

Our integrated train timetabling and locomotive planning model can also be applied to modern high-speed railway systems in which electric multiple units are used. Consider the situation where a set of train services is given; that is, the train route, the departure time window at the origin station, and the arrival time window at the destination station are given. If we view each multiple unit as a "locomotive" and each train service as a "train," then we may use our model to simultaneously assign the multiple units to train services, determine the routes for reallocating the empty multiple units, and determine the timetables for the train services.

Our model is related to the vehicle routing problem with pickup, delivery, and time window constraints (VRPPDTW), where the vehicles and passengers in the problem represent locomotives and trains, respectively. In fact, our model can be viewed as a VRPPDTW with additional constraints, such as track capacity constraints and minimum headway constraints, that exist in train operations. Various solution methods have been proposed for the VRPPDTW. Recently, Mahmoudi and Zhou (2016) developed a state-space-time network representation and a Lagrangian relaxation based solution method for the VRPPDTW. Their solution approach is able to solve the VRPPDTW in large-scale transportation networks. Enlightened by their work, in this paper we also use a state-space-time network representation of our problem and present a Lagrangian relaxation algorithm. However, unlike Mahmoudi and Zhou's network in which a state is represented by an array of binary digits, we use a single integer to represent a state. In addition, our solution method is developed with the consideration of the additional train operations constraints, namely "train service constraints," "arrival and departure headway constraints," and "track capacity constraints." These additional constraints make some of the arcs in the state-space-time network incompatible with each other (i.e., at most one of these arcs can be used). Such *incompatibility constraints* do not exist in the traditional VRPPDTW but are quite common in train timetabling models (see, e.g., Caprara et al. 2002, 2007).

The contribution of this paper is threefold. First, we provide a tool for integrated scheduling of trains and locomotives. Second, our main tool, namely the minimum cost multi-commodity network flow model with incompatible arcs and integer flow restrictions in a state-space-time underlying network, is quite flexible. It is easy to modify the cost parameters to accommodate different train timetabling and locomotive planning requirements. In other words, our solution method can potentially be used for more complicated railway planning problems. Third, our computational results report the significance of integrating train timetabling and locomotive planning decisions.

The rest of this paper is organized as follows. In Section 2, we provide a detailed description of our problem. In Section 3, we present a state-space-time network representation and the corresponding multi-commodity network flow formulation of the problem. In Section 4, we describe our Lagrangian relaxation based solution method. Computational experiments are conducted to test the solution method and to assess the benefits obtained by our integrated solution approach, and the results are reported in Section 5. Several extensions of our model are discussed in Section 6. Concluding remarks are made in Section 7.

2 Problem Description

The problem we study, hereinafter referred to as integrated train timetabling and locomotive assignment problem, aims to simultaneously assign locomotives to trains, determine the locomotives' routes, and determine the trains' timetables. Our problem has the following characteristics: (i) The underlying railway network is a general network, with arcs and vertices representing track segments and stations, respectively. (ii) All the track segments consist of one-way (i.e., mono-directional) tracks, and the capacity of each station is unlimited. (iii) The route of each train is given, but the timetable of each train is to be determined. (iv) The initial and final stations of each locomotive are given, but the route of each locomotive is to be determined. (v) Each train needs only one locomotive, and the locomotive needs to be compatible with the train (e.g., with an operating power no smaller than the power requirement of the train). (vi) After a locomotive has finished serving a train, it may light run to another station to serve another train. (vii) A train may be canceled, and a penalty is incurred when there is a cancellation. (viii) The planning horizon, denoted $[0, T]$, is discretized, and the time units are expressed as integers (e.g., $T = 1440$ if the planning horizon is 24 hours and each time unit is 1 minute).

The train timetabling part of our model follows fairly closely the model of Caprara et al. (2002). For simplicity, our model does not consider some features that may appear in practice. These features include, for example, station capacity constraints, two-way tracks where trains and locomotives may travel in opposite directions, etc. See Section 6 for a discussion of possible extensions of our model.

2.1 Input Data

Table 1 summarizes the input parameters of the problem, where all time-related parameters are integer-valued. Detailed explanations of these input parameters are given as follows.

2.1.1 Railway Network Data

Let $N = (S, E)$ be the given railway network, where S and E are the vertex set and arc set, respectively. Each vertex $i \in S$ represents a station. Each arc $i \to j \in E$ represents the mono-directional track segment from station i to station j, with no intermediate station in between. To ensure safety, at each station $i \in S$, there is a minimum time interval g_i between two consecutive arrivals coming from the same track segment (i.e., arrival headway), and there is a minimum time interval h_i between two consecutive departures entering the same track segment (i.e., departure headway).

Table 1: Summary of input data.

Type of data	Notation	Description
Railway network data	S	set of stations
	E	set of mono-directional track segments $i \to j$, where $i, j \in S$
	g_i	minimum headway between arrivals from the same track segment at station i
	h_i	minimum headway between departures to the same track segment at station i
Train data	K	set of trains
	S_k	set of stations along train k's route, where $S_k \subseteq S$
	s_{kj}	jth station along train k's route, where $s_{kj} \in S_k$
	o_k	origin station of train k (i.e., $o_k = s_{k1}$)
	d_k	destination station of train k (i.e., $d_k = s_{k\lvert S_k\rvert}$)
	$[p_k, p'_k]$	departure time window for train k at station o_k, where $0 \leq p_k \leq p'_k \leq T$
	$[q_k, q'_k]$	arrival time window for train k at station d_k, where $0 \leq q_k \leq q'_k \leq T$
	α^k_{ij}	minimum time for train k to traverse $i \to j$
	β_{ki}	minimum required dwell time of train k at station i, where $i \in S_k$
	Ω_k	set of compatible locomotives for train k
	π_k	penalty for canceling train k
	$\phi_k(\cdot)$	nondecreasing function penalizing train k's shift
	γ_k	unit penalty of train k's stretch
Locomotive data	L	set of locomotives
	$\hat{o}_l$	origin station of locomotive l, where $\hat{o}_l \in S$
	$\hat{d}_l$	destination station of locomotive l, where $\hat{d}_l \in S$
	$\hat{p}_l$	earliest start time of operation of locomotive l, where $0 \leq \hat{p}_l \leq T$
	$\hat{q}_l$	latest completion time of operation of locomotive l, where $0 \leq \hat{q}_l \leq T$
	$\hat{\alpha}^l_{ij}$	time for locomotive l to traverse $i \to j$ when light-running
	c_l	operating cost per unit time for locomotive l when it is running
	c'_l	operating cost per unit time for locomotive l when it is not running
	f_{lk}	fixed cost for assigning locomotive l to train k, where $l \in \Omega_k$
	ϵ^+_{lk}	service time for locomotive l to pick up train k at station o_k, where $l \in \Omega_k$
	ϵ^-_{lk}	service time for locomotive l to drop off train k at station d_k, where $l \in \Omega_k$

2.1.2 Train Data

Let K be the set of trains considered. For each $k \in K$, the input data of train k include: (i) a set $S_k = \{s_{k1}, \ldots, s_{k|S_k|}\} \subseteq S$ of distinct stations that train k visits, where $(s_{k1}, \ldots, s_{k|S_k|})$ is the sequence of stations visited, $o_k = s_{k1}$ is the origin station, and $d_k = s_{k|S_k|}$ is the destination station; (ii) the departure time window $[p_k, p'_k]$ for train k to depart from station o_k, and the arrival time window $[q_k, q'_k]$ for train k to arrive at station d_k; (iii) the minimum possible running time α^k_{ij} that train k needs to traverse track segment $i \to j$, which equals the travel time on $i \to j$ if train k is assigned to its best-matched locomotive; (iv) the minimum dwell time β_{ki} that train k needs to spend at station $i \in S_k$, during which passengers can leave and board the train; (v) the set Ω_k of locomotives that are compatible with train k; and (vi) a penalty π_k which is incurred when train k is canceled.

Besides these input data, an *ideal timetable* is also given for each train k. It is the most desirable timetable for train k if we assume that a compatible locomotive and all track segments are available for train k at any time. We impose the same penalty measurement on the train timetable as Caprara et al. (2002) if a train's actual timetable is different from its ideal timetable. For each $k \in K$, there is a penalty $\phi_k(\nu_k)$ on train k's *shift* ν_k, which is the absolute difference between the departure times from station o_k in the actual and ideal timetables, where $\phi_k(\cdot)$ is a nondecreasing function. For each $k \in K$, there is also a penalty $\gamma_k \mu_k$ on train k's *stretch* μ_k, which is the total running time in the actual timetable minus the total running time in the ideal timetable, where the total running time of train k is the time interval between the departure from station o_k and the arrival at station d_k. We assume that in the ideal timetable, the travel time of train k on each track segment $i \to j$ is α_{ij}^k, and the dwell time of train k at each station i is β_{ki}.

2.1.3 Locomotive Data

Let L be the set of locomotives considered. For each $l \in L$, the input data of locomotive l include: (i) the origin station $\hat{o}_l$ and destination station $\hat{d}_l$ of the locomotive; (ii) the earliest possible start time of operation $\hat{p}_l$ at station $\hat{o}_l$, and the latest allowed completion time of operation $\hat{q}_l$ at station $\hat{d}_l$; (iii) the running time $\hat{\alpha}_{ij}^l$ locomotive l needs to traverse track segment $i \to j$ when the locomotive is light-running; (iv) the operating cost of c_l per unit time for the locomotive to run on a track; and (v) the operating cost of c'_l per unit time for the locomotive to wait at a station during the time period $[\hat{p}_l, \hat{q}_l]$.

Note that locomotive l may be assigned to train k if it is compatible with train k; that is, $l \in \Omega_k$. A fixed cost of f_{lk} is incurred if locomotive l is assigned to train k. If locomotive l is assigned to train k, then the speed of this locomotive-train pair is the smaller of locomotive l's light-running speed and train k's speed. Hence, once this assignment is made, the running time required for this locomotive-train pair to traverse track segment $i \to j$ is $\max\{\hat{\alpha}_{ij}^l, \alpha_{ij}^k\}$, for any $i \to j \in E$. When locomotive l picks up train k at station o_k, a service time of ϵ_{lk}^+ is incurred, and an additional dwell time no shorter than β_{ko_k} is needed before this locomotive-train pair can depart from station o_k. When this locomotive-train pair arrives at station d_k, a dwell time no shorter than β_{kd_k} is needed, and a service time of ϵ_{lk}^- is incurred for locomotive l to drop off train k before the locomotive can leave station d_k.

As mentioned in Section 1, our model can be applied to modern high-speed railway systems. In this case, we set the minimum running time α_{ij}^k for a train service k on any track segment $i \to j$ to 0 so that the running time of train service k on track segment $i \to j$ is equal to $\hat{\alpha}_{ij}^l$ if multiple unit l is assigned to this train service.

2.2 Objective and Constraints

The problem is to assign locomotives to trains, determine a route for each locomotive, and determine a timetable for each train, such that the total cost is minimized. The route for locomotive l has to start at station

$\hat{o}_l$ and terminate at station $\hat{d}_l$, and it may visit a station multiple times if needed. Several types of constraints need to be satisfied.

- Train assignment constraints: Each train k is either assigned one compatible locomotive or canceled. Each locomotive l needs to serve those trains assigned to it one after another. When locomotive l serves a train k, it needs to travel to station o_k $(= s_{k1})$ to pick up the train, traverse the path $s_{k1} \to s_{k2} \to \cdots \to s_{k|S_k|}$ with the train, and drop off the train at station d_k $(= s_{k|S_k|})$.
- Time window constraints: Each locomotive l is initially located at station $\hat{o}_l$ and is available for operation at time $\hat{p}_l$. It needs to complete its operation at station $\hat{d}_l$ no later than $\hat{q}_l$. For each train k assigned to locomotive l, locomotive l needs to arrive at station o_k no later than $p'_k - \epsilon^+_{lk} - \beta_{ko_k}$ to pick up train k, spend at least $\epsilon^+_{lk} + \beta_{ko_k}$ time units at station o_k, and depart from station o_k with train k during the time window $[p_k, p'_k]$. Locomotive l needs to spend at least β_{ki} time units with train k at station i for each $i \in \{s_{kj} \mid j = 2, 3, \ldots, |S_k| - 1\}$. It also needs to arrive at station d_k with train k during the time window $[q_k, q'_k]$ and then spend at least $\beta_{kd_k} + \epsilon^-_{lk}$ time units at station d_k.
- Headway constraints at stations: For each $i \in S$, the arrivals of two locomotives at station i must be at least g_i time units apart if both locomotives are arriving from the same track segment, and the departures of two locomotives from station i must be at least h_i time units apart if both locomotives are departing to the same track segment.
- Track capacity constraints: Each arc in network N represents a mono-directional track segment. Thus, a locomotive/train should be forbidden to overtake another locomotive/train on a track segment.

The total cost of a solution is the sum of (i) total fixed cost of all locomotive-train assignments, (ii) total operating cost of all locomotives, (iii) total cancellation cost of all canceled trains, (iv) total penalty on the shift of all (uncanceled) trains, and (v) total penalty on the stretch of all (uncanceled) trains.

Note that in practice it may be undesirable to cancel a train. In our model, we may set π_k to infinity if we want to disallow the cancellation of train k. Note also that in some applications, the train timetable is repeated daily or weekly, and each locomotive should finish a cycle's operation at a station at which the locomotive can start its next cycle's operation. In Section 6.1, we discuss how our model can be applied to such setting.

2.3 An Example

Consider an example with a railway network $N = (S, E)$ as depicted in Figure 1, in which there are five stations and ten one-way track segments. The set of trains is $K = \{k_1, k_2, k_3\}$, and the set of locomotives is $L = \{l_1, l_2\}$. Both the minimum arrival headway and departure headway at each station are 1 time unit (i.e., $g_i = h_i = 1$ for all $i \in S$). Other input parameter values are provided in Table 2, where the cost data have been omitted.

The given ideal timetable for train k_1 is the timetable where the train departs from its origin station i_1 at time 2, traverses each track segment in minimum time, spends the minimum dwell time at each station along its

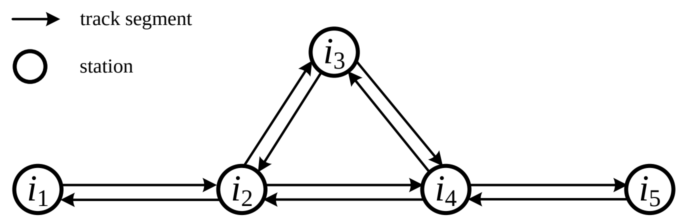


Figure 1: Railway network of the example.

route, and arrives at its destination station i_4 at time 6. The given ideal timetable for train k_2 is the timetable where the train departs from its origin station i_2 at time 12, traverses track segment $i_2 \to i_1$ in minimum time, and arrives at its destination station i_1 at time 13. The given ideal timetable for train k_3 is the timetable where the train departs from its origin station i_5 at time 6, traverses each track segment in minimum time, spends the minimum dwell time at each station along its route, and arrives at its destination station i_1 at time 12. Figure 2(a) depicts these three trains' ideal timetables.

Consider a feasible solution in which locomotive l_1 serves train k_1 first and then train k_2, while locomotive l_2 serves train k_3. Specifically, locomotive l_1 begins its operation at time 0 at its origin station i_1 which is also train k_1's origin station, picks up train k_1 during the time period $[0, 2]$, leaves station i_1 at time 2, visits stations i_2 and i_3, arrives at station i_4 at time 6, and drops off train k_1 at station i_4 during the time period $[6, 8]$. It then light-travels on track segment $i_4 \to i_2$, arrives at station i_2 at time 9, waits for 2 time units, picks up train k_2 at station i_2 during time period $[11, 13]$, leaves station i_2 with train k_2 at time 13, arrives at station i_1 at time 14, drops off train k_2 at station i_1 during the time period $[14, 16]$, and completes its operation at time 16. Since train k_2's departure time at its origin is 1 time unit later than that of its ideal timetable, its shift is 1 time unit. Locomotive l_2 begins its operation at time 4 at its origin station i_5 which is also train k_3's origin station,

Table 2: Input data of the example.

Train k	(o_k, d_k)	$s_{k1} \to s_{k2} \to \cdots \to s_{k\lvert S_k\rvert}$	$(\beta_{k1}, \ldots, \beta_{k\lvert S_k\rvert})$	$[p_k, p'_k]$	$[q_k, q'_k]$	Ω_k
k_1	(i_1, i_4)	$i_1 \to i_2 \to i_3 \to i_4$	$(0, 1, 0, 0)$	$[2, 4]$	$[6, 8]$	$\{l_1\}$
k_2	(i_2, i_1)	$i_2 \to i_1$	$(0, 0)$	$[12, 13]$	$[13, 14]$	$\{l_1\}$
k_3	(i_5, i_1)	$i_5 \to i_4 \to i_2 \to i_1$	$(0, 0, 0, 0)$	$[6, 8]$	$[12, 14]$	$\{l_1, l_2\}$

Locomotive l	$(\hat{o}_l, \hat{d}_l)$	$[\hat{p}_l, \hat{q}_l]$	$(\epsilon^+_{lk_1}, \epsilon^-_{lk_1})$	$(\epsilon^+_{lk_2}, \epsilon^-_{lk_2})$	$(\epsilon^+_{lk_3}, \epsilon^-_{lk_3})$
l_1	(i_1, i_1)	$[0, 16]$	$(2, 2)$	$(2, 2)$	$(2, 2)$
l_2	(i_5, i_1)	$[0, 16]$	-	-	$(2, 2)$

Track segment $i \to j$	$\alpha^{k_1}_{ij}$	$\alpha^{k_2}_{ij}$	$\alpha^{k_3}_{ij}$	$\hat{\alpha}^{l_1}_{ij}$	$\hat{\alpha}^{l_2}_{ij}$
$i_1 \to i_2$	1	-	-	1	2
$i_2 \to i_1$	-	1	2	1	2
$i_2 \to i_3$	1	-	-	1	2
$i_2 \to i_4$	-	-	-	1	2
$i_3 \to i_2$	-	-	-	1	2
$i_3 \to i_4$	1	-	-	1	2
$i_4 \to i_3$	-	-	-	1	2
$i_4 \to i_2$	-	-	2	1	2
$i_4 \to i_5$	-	-	-	1	2
$i_5 \to i_4$	-	-	2	1	2

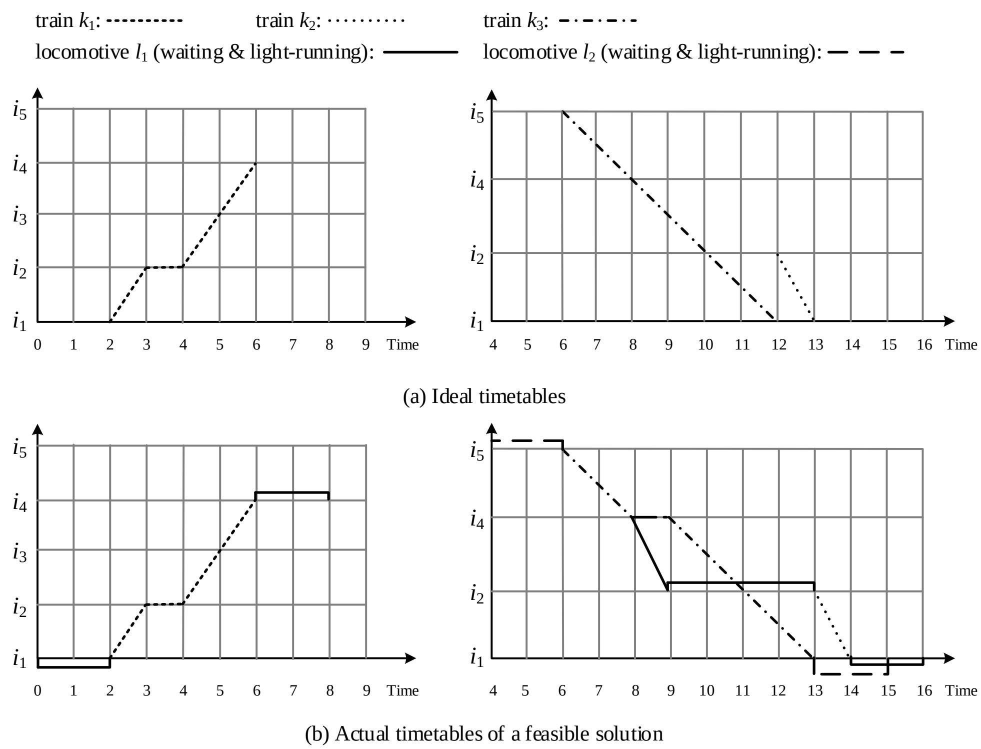


Figure 2: Ideal and actual timetables of the trains.

picks up train k_3 during the time period $[4, 6]$, leave station i_5 at time 6, arrives at station i_4 at time 8, and stays at station i_4 during the time period $[8, 9]$, which is longer than the minimum required dwell time of train k_3 by 1 time unit. It then leaves station i_4 at time 9, visits station i_2, arrives at station i_1 at time 13, drops off train k_3 at station i_1 during the time period $[13, 15]$, and completes its operation at time 15. Since train k_3's running time is 1 time unit longer than that of its ideal timetable, its stretch is 1 time unit. The total cost of this feasible solution is $(c_{l_1} \cdot 13 + c_{l_2} \cdot 10) + (c'_{l_1} \cdot 3 + c'_{l_2} \cdot 1) + (f_{l_1 k_1} + f_{l_1 k_2} + f_{l_2 k_3}) + (\phi_{k_2}(1) + \gamma_{k_3} \cdot 1)$, where the first part is the operating cost of the locomotives running on tracks, the second part is the operating cost of the locomotives waiting at stations, the third part is the fixed cost of locomotive-train assignment, and the fourth part is the penalty on the train services. The train cancellation penalty is zero as no train is canceled. Figure 2(b) depicts the three trains' actual timetables, as well as the schedules and routes of the locomotives, of this feasible solution.

Note that in this feasible solution train k_3 dwells at station i_4 longer than the minimum required dwell time. This is because train k_3 is ready to leave station i_4 at time 8 and head toward station i_2, but locomotive l_1 is also departing from station i_4 at time 8 and heading toward station i_2. The departure headway requirement demands these two departures to be at least one time unit apart. Note also that train k_2 departs from its origin station i_2 later than that of its ideal timetable. This is because if train k_2 departs from its origin station at its ideal timetable's departure time (i.e., time 12), then it will arrive at station i_1 at the same time as train k_3 (i.e., time 13). The arrival headway requirement demands these two arrivals to be at least one time unit apart.

3 The State-Space-Time Network Formulation

In this section we formulate our problem as a minimum cost multi-commodity network flow problem with incompatible arcs and integer flow restrictions, where each commodity represents a locomotive. The underlying network is a three-dimensional acyclic directed network $G = (V, A)$, where the three dimensions are state, space, and time.

3.1 State-Space-Time Network Construction

For each locomotive, we use a *state* to indicate which train the locomotive is serving. We say that a locomotive is in state 0 if the locomotive is not serving any train (i.e., if the locomotive is light-running or staying at a station by itself). We say that a locomotive is in state k if it is serving train k (i.e., if the locomotive is picking up train k, pulling train k, or dropping off train k), for $k \in K$. Let $\bar{K} = \{0\} \cup K$ denote the set of all possible states, which forms the "state" dimension of network G. The set of all possible time instants in the planning horizon is $\{0, 1, \ldots, T\}$, which forms the "time" dimension of G. The "space" dimension of G covers the railway stations, where we use two vertices to represent one station for each state and time combination. One of these two vertices corresponds to the arrival at the station, and the other corresponds to the departure from the station. We use these two vertices because each train has a dwell time requirement at a station; see Caprara et al. (2002) for a similar treatment. Mathematically, we let $\bar{S} = \{\rho(i) \mid i \in S\} \cup \{\rho'(i) \mid i \in S\}$ denote the "space" dimension, where $\rho(i)$ represents a locomotive's arrival at station i and $\rho'(i)$ represents a locomotive's departure from station i.

The vertex set of the state-space-time network G is

$$V = \{\bar{o}, \bar{d}\} \cup \big\{(k, r, t) \bigm| k \in \bar{K},\, r \in \bar{S},\, t = 0, 1, \ldots, T\big\},$$

where vertices $\bar{o}$ and $\bar{d}$ are the dummy source and dummy sink, respectively, for the multi-commodity flow. A path in this network starting at $\bar{o}$ and ending at $\bar{d}$ indicates the sequence of changes in state and location of a locomotive over time. The arc set A of network G contains several types of arcs, with a vector of cost coefficients $(\xi^{l_1}_{u,v}, \ldots, \xi^{l_{|L|}}_{u,v})$ associated with each arc $u \to v \in A$, where cost coefficient $\xi^l_{u,v}$ represents the cost for locomotive l to traverse this arc. Each arc $u \to v \in A$ has a unit capacity per locomotive, but different locomotives may traverse the same arc. Descriptions of different types of arcs are given below.

- Starting arcs: For each $i \in S$ and $t = 0, 1, \ldots, T$, there is a starting arc $\bar{o} \to (0, \rho(i), t)$ if there exists at least one locomotive $l \in L$ such that $i = \hat{o}_l$ and $t \geq \hat{p}_l$. For each $l \in L$, if $i = \hat{o}_l$ and $t \geq \hat{p}_l$, then $\xi^l_{\bar{o},(0,\rho(i),t)} = 0$; otherwise, $\xi^l_{\bar{o},(0,\rho(i),t)} = +\infty$. This arc allows locomotive l to start its operation at station $\hat{o}_l$ at or after time $\hat{p}_l$.
- Ending arcs: For each $i \in S$ and $t = 0, 1, \ldots, T$, there is an ending arc $(0, \rho'(i), t) \to \bar{d}$ if there exists at least one locomotive $l \in L$ such that $i = \hat{d}_l$ and $t \geq \hat{p}_l$. For each $l \in L$, if $i = \hat{d}_l$ and $t \geq \hat{p}_l$, then $\xi^l_{(0,\rho'(i),t),\bar{d}} = 0$; otherwise, $\xi^l_{(0,\rho'(i),t),\bar{d}} = +\infty$. This arc allows locomotive l to complete its operation at station $\hat{d}_l$ at or before time $\hat{q}_l$.

- Pickup arcs: For each $k \in K$ and $t, t' = 0, 1, \ldots, T$, there is a pickup arc $(0, \rho'(o_k), t) \to (k, \rho(o_k), t')$ if there exists at least one locomotive $l \in \Omega_k$ such that $t \geq \hat{p}_l$ and $t' = t + \epsilon^+_{lk} \leq \hat{q}_l$. For each $l \in L$, if $l \in \Omega_k$, $t \geq \hat{p}_l$, and $t' = t + \epsilon^+_{lk} \leq \hat{q}_l$, then $\xi^l_{(0,\rho'(o_k),t),(k,\rho(o_k),t')} = f_{lk} + c'_l(t' - t) - \pi_k$; otherwise, $\xi^l_{(0,\rho'(o_k),t),(k,\rho(o_k),t')} = +\infty$. This arc allows locomotive l to pick up train k at station o_k and change the state from 0 to k. The cost coefficient $\xi^l_{(0,\rho'(o_k),t),(k,\rho(o_k),t')}$ includes not only the locomotive's operating cost $c'_l(t' - t)$ but also the locomotive-train assignment cost f_{lk}. Since train k is picked up by locomotive l, the train is not canceled. Hence, a cost reduction of π_k is also included in the cost coefficient $\xi^l_{(0,\rho'(o_k),t),(k,\rho(o_k),t')}$. Note that this arc changes the "space" component from $\rho'(o_k)$ to $\rho(o_k)$. This allows locomotive l to wait at station o_k (which changes the status from "arrival" to "departure") before picking up train k, and allows train k to dwell at station o_k for passenger boarding (which also changes the status from "arrival" to "departure") after being picked up by locomotive l; see the descriptions of dwelling arcs and waiting arcs below.

- Drop-off arcs: For each $k \in K$ and $t, t' = 0, 1, \ldots, T$, there is a drop-off arc $(k, \rho'(d_k), t) \to (0, \rho(d_k), t')$ if there exists at least one locomotive $l \in \Omega_k$ such that $t \geq \hat{p}_l$ and $t' = t + \epsilon^-_{lk} \leq \hat{q}_l$. For each $l \in L$, if $l \in \Omega_k$, $t \geq \hat{p}_l$, and $t' = t + \epsilon^-_{lk} \leq \hat{q}_l$, then $\xi^l_{(k,\rho'(d_k),t),(0,\rho(d_k),t')} = c'_l(t' - t)$; otherwise, $\xi^l_{(k,\rho'(d_k),t),(0,\rho(d_k),t')} = +\infty$. This arc allows locomotive l to drop off train k at station d_k and change the state from k to 0. Similar to pickup arcs, this arc changes the "space" component from $\rho'(d_k)$ to $\rho(d_k)$.

- Light-running arcs: For each $i \to j \in E$ and $t, t' = 0, 1, \ldots, T$, there is a light-running arc $(0, \rho'(i), t) \to (0, \rho(j), t')$ if there exists at least one locomotive $l \in L$ such that $t \geq \hat{p}_l$, $t' = t + \hat{\alpha}^l_{ij}$, and $t' \leq \hat{q}_l$. For each $l \in L$, if $t \geq \hat{p}_l$, $t' = t + \hat{\alpha}^l_{ij}$, and $t' \leq \hat{q}_l$, then $\xi^l_{(0,\rho'(i),t),(0,\rho(j),t')} = c_l(t' - t)$; otherwise, $\xi^l_{(0,\rho'(i),t),(0,\rho(j),t')} = +\infty$. This arc allows locomotive l to light run along track segment $i \to j$.

- Loaded-running arcs: For each $k \in K$, $i \to j \in E$, and $t, t' = 0, 1, \ldots, T$, there is a loaded-running arc $(k, \rho'(i), t) \to (k, \rho(j), t')$ if there exists at least one locomotive $l \in \Omega_k$ such that:
  (C1) $t \geq \hat{p}_l$;
  (C2) $t' = t + \max\{\hat{\alpha}^l_{ij}, \alpha^k_{ij}\} \leq \hat{q}_l$;
  (C3) track segment $i \to j$ is along the path $s_{k1} \to s_{k2} \to \cdots \to s_{k|S_k|}$;
  (C4) $p_k \leq t \leq p'_k$ if $i = s_{k1}$; and
  (C5) $q_k \leq t' \leq q'_k$ if $j = s_{k|S_k|}$.
  Let τ^k denote the departure time at station o_k in train k's ideal timetable. For each $l \in L$, if $l \in \Omega_k$, conditions (C1)–(C5) are satisfied, and track segment $i \to j$ is along the path $s_{k2} \to s_{k3} \to \cdots \to s_{k|S_k|}$, then $\xi^l_{(k,\rho'(i),t),(k,\rho(j),t')} = c_l(t' - t) + \gamma_k(t' - t - \alpha^k_{ij})$. If $l \in \Omega_k$, conditions (C1)–(C5) are satisfied, and track segment $i \to j$ is $s_{k1} \to s_{k2}$, then $\xi^l_{(k,\rho'(i),t),(k,\rho(j),t')} = c_l(t' - t) + \gamma_k(t' - t - \alpha^k_{ij}) + \phi_k(|t - \tau^k|)$. Otherwise, $\xi^l_{(k,\rho'(i),t),(k,\rho(j),t')} = +\infty$. This arc allows train k to travel along track segment $i \to j$ during the time period $[t, t']$. The cost coefficient $\xi^l_{(k,\rho'(i),t),(k,\rho(j),t')}$ includes locomotive l's operating cost as well as the penalty of train k's stretch contributed by the traveling along track segment $i \to j$. If station i is the origin station of train k, then the penalty of train k's shift is also included in $\xi^l_{(k,\rho'(i),t),(k,\rho(j),t')}$.

- Dwelling arcs: For each $k \in K$, $i \in S$, and $t, t' = 0, 1, \ldots, T$, there is a dwelling arc $(k, \rho(i), t) \to (k, \rho'(i), t')$ if there exists $l \in \Omega_k$ such that $t \geq \hat{p}_l$, $t' = t + \beta_{ki} \leq \hat{q}_l$, and $i \in \{s_{k1}, \ldots, s_{k|S_k|}\}$. For each $l \in L$, if $l \in \Omega_k$, $t \geq \hat{p}_l$, $t' = t + \beta_{ki} \leq \hat{q}_l$, and $i \in \{s_{k1}, \ldots, s_{k|S_k|}\}$, then $\xi^l_{(k,\rho(i),t),(k,\rho'(i),t')} = c'_l(t' - t) + \gamma_k(t' - t - \beta_{ki})$; otherwise, $\xi^l_{(k,\rho(i),t),(k,\rho'(i),t')} = +\infty$. This arc allows train k to spend a minimum required dwell time at station i, during which passengers can alight and board the train. For each $k \in K$, $i \in S$, and $t = 0, 1, \ldots, T - 1$, there is also a dwelling arc $(k, \rho'(i), t) \to (k, \rho'(i), t + 1)$. For each $l \in L$, if $l \in \Omega_k$, $i \in S$, and $t = 0, 1, \ldots, T - 1$, then $\xi^l_{(k,\rho'(i),t),(k,\rho'(i),t+1)} = c'_l + \gamma_k$; otherwise, $\xi^l_{(k,\rho'(i),t),(k,\rho'(i),t+1)} = +\infty$. This arc allows train k to continue to stay at station i after it has spent its minimum required dwell time there. The cost coefficients $\xi^l_{(k,\rho(i),t),(k,\rho'(i),t')}$ and $\xi^l_{(k,\rho'(i),t),(k,\rho'(i),t+1)}$ include locomotive l's operating cost as well as the penalty of train k's stretch contributed by the dwelling at station i.
- Waiting arcs: For each $i \in S$ and $t = 0, 1, \ldots, T - 1$, there is a waiting arc $(0, \rho'(i), t) \to (0, \rho'(i), t + 1)$ if there exists $l \in L$ such that $\hat{p}_l \leq t \leq \hat{q}_l - 1$. For each $l \in L$, if $\hat{p}_l \leq t \leq \hat{q}_l - 1$, then $\xi^l_{(0,\rho'(i),t),(0,\rho'(i),t+1)} = c'_l$; otherwise, $\xi^l_{(0,\rho'(i),t),(0,\rho'(i),t+1)} = +\infty$. This arc allows a locomotive to wait at station i when it is not serving any train. For each $i \in S$ and $t = 0, 1, \ldots, T$, there is also a waiting arc $(0, \rho(i), t) \to (0, \rho'(i), t)$ if there exists $l \in L$ such that $\hat{p}_l \leq t \leq \hat{q}_l$. For each $l \in L$, if $\hat{p}_l \leq t \leq \hat{q}_l$, then $\xi^l_{(0,\rho(i),t),(0,\rho'(i),t)} = 0$; otherwise, $\xi^l_{(0,\rho(i),t),(0,\rho'(i),t)} = +\infty$. This arc allows a locomotive to switch from "arrival" to "departure" at station i when the locomotive is not serving any train.

A path from vertex $\bar{o}$ to vertex $\bar{d}$ in this state-space-time network corresponds to a possible schedule of a locomotive, and this schedule is feasible for locomotive l if and only if all the cost coefficients of commodity l along this path are finite. Given this network, we would like to determine one unit of flow for each of the $|L|$ commodities from vertex $\bar{o}$ to vertex $\bar{d}$, such that some incompatibility constraints are satisfied, and that the total cost is minimized. Descriptions of these incompatibility constraints are provided in Section 3.2.

Consider the example and its feasible solution discussed in Section 2.3. Figure 3 depicts locomotive l_1's path in network G. The path begins at vertex $\bar{o}$, traverses the starting arc $\bar{o} \to (0, \rho(i_1), 0)$, and traverses the waiting arc $(0, \rho(i_1), 0) \to (0, \rho'(i_1), 0)$. Then, the path traverses the pickup arc $(0, \rho'(i_1), 0) \to (k_1, \rho(i_1), 2)$, which corresponds to the pickup of train k_1 by locomotive l_1. Next, the path traverses a series of dwelling and loaded-running arcs until it reaches vertex $(k_1, \rho'(i_4), 6)$. Then, the path traverses the drop-off arc $(k_1, \rho'(i_4), 6) \to (0, \rho(i_4), 8)$, which corresponds to the drop-off of train k_1. Next, the path traverses a series of waiting and light-running arcs until it reaches vertex $(0, \rho'(i_2), 11)$. Then, the path traverses the pickup arc $(0, \rho'(i_2), 11) \to (k_2, \rho(i_2), 13)$, which corresponds to the pickup of train k_2, and so on.

Note that if all time-related parameters are positive, then besides the starting and ending arcs, each arc of network G is either of the form $(*, *, t) \to (*, *, t')$ with $t' > t$, or of the form $(*, \rho(i), t) \to (*, \rho'(i), t)$. Thus, network G is acyclic when all time-related parameters are positive. Note also that network G is very large. However, the size of G can be reduced substantially by removing (i) all arcs of which all $|L|$ components of the cost coefficient vector are infinite; (ii) all non-dummy vertices with no incoming arcs as well as the outgoing

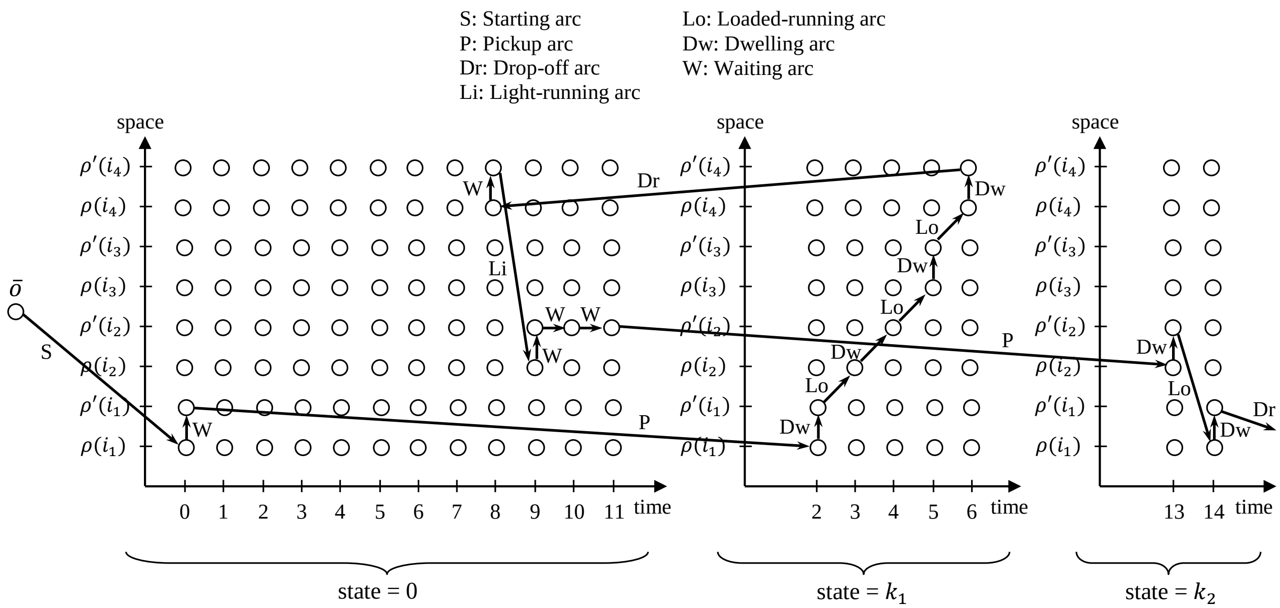


Figure 3: Locomotive l_1's path in the state-space-time network.

arcs of these vertices; and (iii) all non-dummy vertices with no outgoing arcs as well as the incoming arcs of these vertices.

3.2 Incompatibility Constraints

The state-space-time network G is constructed in such a way that any feasible flow with finite cost in this network must satisfy the time window constraints described in Section 2.2. However, a flow in this network may not satisfy the headway constraints and track capacity constraints. In addition, a flow in this network may have some trains being served multiple times. Hence, besides the standard network flow constraints such as flow balance constraints, supply/demand constraints, and capacity constraints, our multi-commodity flow model also has the following incompatibility constraints:

- Train service constraints: Each time a locomotive picks up a train k, a cancellation penalty of π_k can be saved. For each $k \in K$, since train k should be served at most once (and the penalty π_k can be saved at most once), we impose a constraint that the total flow along the pickup arcs in the arc subset

$$C_k^1 = A \cap \left\{(0, \rho'(o_k), t) \to (k, \rho(o_k), t') \;\middle|\; t, t' = 0, 1, \ldots, T\right\}$$

  is at most one.

- Arrival headway constraints: The time difference between two arrivals at the same station j should be no less than the required arrival headway g_j if both arrivals are arriving from the same station i. Hence, for each $i \to j \in E$ and each $t_1 = 0, 1, \ldots, T - g_j + 1$, no more than one locomotive may finish traversing track segment $i \to j$ during the time interval $[t_1, t_1 + g_j - 1]$. Thus, for each $i \to j \in E$ and each $t_1 = 0, 1, \ldots, T - g_j + 1$, we impose a constraint that the total flow along the light-running and loaded-running arcs in the arc subset

$$C^2_{ijt_1} = A \cap \big\{(k, \rho'(i), t) \to (k, \rho(j), t') \;\big|\; k \in \bar{K};\ t, t' = 0, 1, \ldots, T;\ t_1 \le t' \le t_1 + g_j - 1\big\}$$

is at most one.

- Departure headway constraints: The time difference between two departures from the same station i should be no less than the required departure headway h_i if both departures are leaving for the same station j. Hence, for each $i \to j \in E$ and each $t_1 = 0, 1, \ldots, T - h_i + 1$, no more than one locomotive may start traversing track segment $i \to j$ during the time interval $[t_1, t_1 + h_i - 1]$. Thus, for each $i \to j \in E$ and each $t_1 = 0, 1, \ldots, T - h_i + 1$, we impose a constraint that the total flow along the light-running and loaded-running arcs in the arc subset

$$C^3_{ijt_1} = A \cap \big\{(k, \rho'(i), t) \to (k, \rho(j), t') \;\big|\; k \in \bar{K};\ t, t' = 0, 1, \ldots, T;\ t_1 \le t \le t_1 + h_i - 1\big\}$$

is at most one.

- Track capacity constraints: A locomotive is not allowed to overtake another locomotive/train when traveling on a track segment. Thus, for each $i \to j \in E$ and each group of four distinct time points $0 \le t_1 < t_2 < t_3 < t_4 \le T$, we impose a constraint that the total flow along the light-running and loaded-running arcs in the arc subset

$$C^4_{ijt_1t_2t_3t_4} = A \cap \Big[\big\{(k, \rho'(i), t_1) \to (k, \rho(j), t_4) \;\big|\; k \in \bar{K}\big\} \cup \big\{(k, \rho'(i), t_2) \to (k, \rho(j), t_3) \;\big|\; k \in \bar{K}\big\}\Big]$$

is at most one.

Denote

$$\begin{aligned}\mathcal{C} = &\{C^1_k \mid k \in K\} \cup \{C^2_{ijt_1} \mid i \to j \in E;\ t_1 = 0, 1, \ldots, T - g_j + 1\} \\ &\cup \{C^3_{ijt_1} \mid i \to j \in E;\ t_1 = 0, 1, \ldots, T - h_i + 1\} \cup \{C^4_{ijt_1t_2t_3t_4} \mid i \to j \in E;\ 0 \le t_1 < t_2 < t_3 < t_4 \le T\},\end{aligned}$$

which is a collection of all the sets of incompatible arcs in network G. For any $C \in \mathcal{C}$, the total flow along the arcs in C cannot exceed one.

3.3 Integer Programming Formulation

For $l \in L$ and $u \to v \in A$, let $x^l_{u,v} = 1$ if arc $u \to v$ is on the path traversed by locomotive l; and let $x^l_{u,v} = 0$ otherwise. Then, the above minimum cost multi-commodity network flow problem with incompatible arcs and integer flow restrictions can be formulated as the following integer program.

$$\begin{aligned}
\mathbf{P}: \ \text{Minimize} \quad & \textstyle\sum_{k \in K} \pi_k + \sum_{l \in L} \sum_{u \to v \in A} \xi^l_{u,v} x^l_{u,v} && && (1)\\
\text{subject to} \quad & \textstyle\sum_{\{v : \bar{o} \to v \in A\}} x^l_{\bar{o},v} = 1, && \text{for all } l \in L && (2)\\
& \textstyle\sum_{\{u : u \to \bar{d} \in A\}} x^l_{u,\bar{d}} = 1, && \text{for all } l \in L && (3)\\
& \textstyle\sum_{\{u : u \to v \in A\}} x^l_{u,v} = \sum_{\{w : v \to w \in A\}} x^l_{v,w}, && \text{for all } l \in L;\ v \in V \setminus \{\bar{o}, \bar{d}\} && (4)
\end{aligned}$$

$$\textstyle\sum_{u\to v\in C}\sum_{l\in L} x^l_{u,v} \le 1, \qquad \text{for all } C\in\mathcal{C} \tag{5}$$

$$x^l_{u,v}\in\{0,1\}, \qquad \text{for all } l\in L;\ u\to v\in A \tag{6}$$

In objective function (1), the constant term "$\sum_{k\in K}\pi_k$" is the total cancellation penalty of all the trains. The term "$\sum_{l\in L}\sum_{u\to v\in A}\xi^l_{u,v}x^l_{u,v}$" is the total cost of the solution minus the cancellation penalties saved by successfully assigning compatible locomotives to trains. Thus, objective function (1) equals the total cost of the solution. Constraints (2), (3), and (4) are the supply, demand, and flow balance constraints, respectively, in a standard multi-commodity network flow model. Constraints (5) cover all the incompatibility constraints presented in Section 3.2. These are also known as "clique constraints"; see Caprara et al. (2002). Constraints (6) are the binary constraints of the decision variables.

4 Lagrangian Relaxation Heuristic

In this section, we present a Lagrangian relaxation heuristic for solving problem **P**. Lagrangian relaxation is a widely used method in solving train routing and timetabling problems; see, for example, Brännlund et al. (1998), Cacchiani et al. (2010a, 2010b, 2012), Meng and Zhou (2014), and Zhou and Teng (2016).

4.1 The Lagrangian Relaxation

Relaxing constraints (5) of problem **P** and bringing them into the objective function with associated Lagrangian multipliers $\lambda_C \ge 0$ ($C\in\mathcal{C}$), we obtain the following relaxed problem, where λ denotes the vector of the λ_C values.

$$\begin{aligned}
\mathbf{P}(\lambda):\ \text{Minimize}\quad & \textstyle\sum_{k\in K}\pi_k+\sum_{l\in L}\sum_{u\to v\in A}\xi^l_{u,v}x^l_{u,v}+\sum_{C\in\mathcal{C}}\lambda_C(\sum_{u\to v\in C}\sum_{l\in L}x^l_{u,v}-1)\\
\text{subject to}\quad & \textstyle\sum_{\{v:\bar{o}\to v\in A\}} x^l_{\bar{o},v}=1, && \text{for all } l\in L\\
& \textstyle\sum_{\{u:u\to\bar{d}\in A\}} x^l_{u,\bar{d}}=1, && \text{for all } l\in L\\
& \textstyle\sum_{\{u:u\to v\in A\}} x^l_{u,v}=\sum_{\{w:v\to w\in A\}} x^l_{v,w}, && \text{for all } l\in L;\ v\in V\setminus\{\bar{o},\bar{d}\}\\
& x^l_{u,v}\in\{0,1\}, && \text{for all } l\in L;\ u\to v\in A
\end{aligned}$$

After removing the constant $\sum_{k\in K}\pi_k-\sum_{C\in\mathcal{C}}\lambda_C$ from the objective function, this Lagrangian relaxation problem can be decomposed into $|L|$ independent subproblems. The subproblem corresponding to each $l\in L$ is given as follows.

$$\begin{aligned}
\mathbf{P}_l(\lambda):\ \text{Minimize}\quad & \textstyle\sum_{u\to v\in A}\xi^l_{u,v}x^l_{u,v}+\sum_{C\in\mathcal{C}}\lambda_C\sum_{u\to v\in C}x^l_{u,v}\\
\text{subject to}\quad & \textstyle\sum_{\{v:\bar{o}\to v\in A\}} x^l_{\bar{o},v}=1,\\
& \textstyle\sum_{\{u:u\to\bar{d}\in A\}} x^l_{u,\bar{d}}=1,\\
& \textstyle\sum_{\{u:u\to v\in A\}} x^l_{u,v}=\sum_{\{w:v\to w\in A\}} x^l_{v,w}, && \text{for all } v\in V\setminus\{\bar{o},\bar{d}\}\\
& x^l_{u,v}\in\{0,1\}, && \text{for all } u\to v\in A
\end{aligned}$$

Each subproblem $\mathbf{P}_l(\lambda)$ is a shortest path problem with arc length $\delta^l_{u,v} = \xi^l_{u,v} + \sum_{\{C\in\mathcal{C}:u\to v\in C\}} \lambda_C$. This shortest path problem in the acyclic network G can be solved efficiently using a standard dynamic programming algorithm. Furthermore, subproblems $\mathbf{P}_l(\lambda)$ can be solved in parallel for different $l \in L$ when they are solved by a multicore computer processor.

4.2 The Subgradient Optimization Procedure

For any vector λ of nonnegative λ_C values, the optimal objective function value of $\mathbf{P}(\lambda)$ is a lower bound on the optimal objective value of problem $\mathbf{P}$. Near-optimal λ_C values can be obtained by a subgradient optimization procedure. Because the number of relaxed constraints is very large, we use the *dynamic constraint-generation scheme* similar to that in Caprara et al. (2002) to handle the relaxed constraints and generate the λ_C values. Under this scheme, we dynamically identify constraints that the relaxed solution violates, and we use a constraint pool to store these constraints.

Note that the violated constraints include four types of incompatibility constraints described in Section 3.2. The number of train service constraints is relatively small. Thus, we include all train service constraints in the constraint pool (i.e., the constraint pool initially contains all the train service constraints but none of the other incompatibility constraints). We initialize all λ_C values to zero. At each iteration of the subgradient optimization procedure, we identify arrival headway constraints, departure headway constraints, and track capacity constraints that the relaxed solution violates. Specifically, at each iteration, we consider each pair of locomotives $l, l' \in L$ and their paths in the relaxed solution. Whenever these two paths violate the arrival headway constraint, we include the corresponding constraint in the pool. That is, when these two locomotives traverse track segment $i \to j$ and arrive at station $j \in S$ at time t and t' respectively, where $t \le t' \le t + g_j - 1$, we add the violated constraint corresponding to set C^2_{ijt} to the pool if that constraint is not yet in the pool. Whenever these two paths violate the departure headway constraint, we include the corresponding constraint in the pool. That is, when these two locomotives depart from station $i \in S$ at time t and t' respectively and traverse track segment $i \to j$, where $t \le t' \le t + h_i - 1$, we add the violated constraint corresponding to set C^3_{ijt} to the pool if that constraint is not yet in the pool. Whenever these two paths violate the track capacity constraint, we include the corresponding constraint in the pool. That is, when these two locomotives traverse track segment $i \to j$ during the time intervals $[t_1, t_4]$ and $[t_2, t_3]$ respectively, where $t_1 < t_2 < t_3 < t_4$, we add the violated constraint corresponding to set $C^4_{ijt_1t_2t_3t_4}$ to the pool if that constraint is not yet in the pool.

After adding these violated constraints to the constraint pool, we update the Lagrangian multipliers of the constraints in the pool as follows. Consider the mth constraint in the pool. Let λ^m denote the Lagrangian multiplier corresponding to this constraint, and η^m denote the left-hand side value minus the right-hand side value of this constraint in the relaxed solution. Thus, η^m is the mth component of the subgradient vector η corresponding to the current relaxed solution. For each m, the Lagrangian multiplier λ^m can be updated according to the formula (Held and Karp 1971):

$$\lambda^m \leftarrow \max\Big\{\lambda^m + \theta \cdot \frac{UB - LB(\lambda)}{\|\eta\|^2} \cdot \eta^m,\, 0\Big\},$$

where UB is the value of the best feasible solution of problem $\mathbf{P}$ identified so far, $LB(\lambda)$ is the optimal objective value of $\mathbf{P}(\lambda)$ corresponding to the current Lagrangian multiplier values, and θ is a prespecified step size parameter.

To improve the convergence of the procedure, we further apply a *modified subgradient technique* proposed by Camerini et al. (1975). We use a modified subgradient vector $\tilde{\eta}$ instead of η to update the Lagrangian multipliers λ^m. In each iteration of the procedure, the modified subgradient vector $\tilde{\eta}$ is updated by

$$\tilde{\eta} \leftarrow \eta + b\tilde{\eta},$$

where b is a scalar defined as

$$b = \begin{cases} -a \cdot \frac{\tilde{\eta} \cdot \eta}{\|\tilde{\eta}\|^2}, & \text{if } \tilde{\eta} \cdot \eta < 0; \\ 0, & \text{otherwise;} \end{cases}$$

and a is a prespecified value between 0 and 2. This modified subgradient technique tends to avoid "zig-zag" behavior of the λ^m values caused by subgradient vector η, and thus improves the convergence of the subgradient optimization procedure (see Camerini et al. 1975, Sec. 2).

Each iteration of the subgradient optimization procedure comprises the following steps: (i) solve problem $\mathbf{P}(\lambda)$ using dynamic programming to obtain a lower bound on problem $\mathbf{P}$; (ii) obtain a feasible solution of problem $\mathbf{P}$ using an upper bound heuristic (see Section 4.3 for details); (iii) identify constraints that the current lower bound solution has violated; (iv) update the modified subgradient vector; and (v) update the Lagrangian multipliers. However, in some iterations, step (ii) may be skipped in order to save computational time (see Section 5.1 for details). This procedure is terminated when either the number of iterations reaches a prespecified limit, or the gap between the upper bound and lower bound is less than a prespecified threshold.

4.3 Upper Bound Heuristic

We now present an upper bound heuristic that returns a feasible solution of problem $\mathbf{P}$ for any given vector λ of Lagrangian multipliers $\lambda_C \geq 0$ for $C \in \mathcal{C}$. This heuristic is used in the subgradient optimization procedure presented in Section 4.2 and contains the following two phases; see Algorithm 1 for a summary of this heuristic.

In Phase 1, the heuristic constructs a feasible solution to problem $\mathbf{P}$ based on the arc lengths $\delta^l_{u,v} = \xi^l_{u,v} + \sum_{\{C \in \mathcal{C}: u \to v \in C\}} \lambda_C$, $u \to v \in A$, defined in Section 4.1. For each $l \in L$, the optimal objective value of subproblem $\mathbf{P}_l(\lambda)$ represents the contribution of locomotive l to the optimal objective value of the relaxed problem $\mathbf{P}(\lambda)$. Thus, the heuristic first ranks the locomotives $l \in L$ in nondecreasing order of the optimal objective values of $\mathbf{P}_l(\lambda)$ before constructing a schedule for each locomotive one by one according to this order. The constructed schedule for each locomotive corresponds to a path from vertex $\bar{o}$ to vertex $\bar{d}$ in network G. When the heuristic constructs a schedule for a locomotive $l \in L$, the schedules of those locomotives with higher

Algorithm 1 Upper bound heuristic for constructing schedules for the locomotives in L

Phase 1: Construct initial schedules
Rank the locomotives in L in nondecreasing order of their optimal objective values of $\mathbf{P}_l(\lambda)$, and denote the ranked order of locomotives as $l_1, l_2, \dots, l_{|L|}$;
for $l := l_1, l_2, \dots, l_{|L|}$ **do**
    apply the restricted labeling algorithm to construct a schedule for l, where the schedules of the locomotives with higher ranks than l are kept unchanged
end for
Phase 2: Construct improved schedules
Rearrange $l_1, l_2, \dots, l_{|L|}$ in a random order;
for $l := l_1, l_2, ..., l_{|L|}$ **do**
    apply the restricted labeling algorithm to construct a new schedule for l, where the schedules of all other locomotives are kept unchanged;
    if the new schedule of l is better than the schedule of l obtained in Phase 1, then assign the new schedule to l
end for

ranks than l are already determined and will remain unchanged. To ensure that the schedule for locomotive l does not violate any incompatibility constraints of problem $\mathbf{P}$, we need to prevent its corresponding path in network G from traversing the following arcs, where arc subsets C_k^1, $C_{ijt_1}^2$, $C_{i,j,t_1-g_j+1}^2$, $C_{ijt_1}^3$, $C_{i,j,t_1-h_i+1}^3$, and $C_{ijt_1t_2t_3t_4}^4$ are defined in Section 3.2:

- Due to the train service constraints, for each $k \in K$, if train k has been picked up by one of the locomotives of higher ranks than l, then no pickup arc in the arc subset C_k^1 can be traversed by l.
- Due to the arrival headway constraints, for each $i \to j \in E$ and each $t_1 = 0, 1, ..., T - g_j + 1$, if one of the locomotives of higher ranks than l traverses track segment $i \to j$ and arrives at station j at time t_1, then no light-running or loaded-running arc in the arc subset $C_{ijt_1}^2 \cup C_{i,j,t_1-g_j+1}^2$ can be traversed by l.
- Due to the departure headway constraints, for each $i \to j \in E$ and each $t_1 = 0, 1, ..., T - h_i + 1$, if one of the locomotives of higher ranks than l departs from station i at time t_1 and traverses track segment $i \to j$, then no light-running or loaded-running arc in the arc subset $C_{ijt_1}^3 \cup C_{i,j,t_1-h_i+1}^3$ can be traversed by l.
- Due to the track capacity constraints, for each $i \to j \in E$ and each group of four distinct time points $0 \le t_1 < t_2 < t_3 < t_4 \le T$, if one of the locomotives of higher ranks than l traverses track segment $i \to j$ during time period $[t_1, t_4]$ or time period $[t_2, t_3]$, then no light-running or loaded-running arc in the arc subset $C_{ijt_1t_2t_3t_4}^4$ can be traversed by l.

Let $\bar{C}_1^l$ be the collection of these arcs. In order to ensure that no arc in $\bar{C}_1^l$ is traversed by locomotive l, we revise their arc lengths to $+\infty$. Letting $\chi_{u,v}^l$ denote the revised arc length of $u \to v \in A$, we have

$$\chi_{u,v}^l = \begin{cases} +\infty, & \text{if } u \to v \in \bar{C}_1^l; \\ \delta_{u,v}^l, & \text{otherwise.} \end{cases}$$

Using the arc lengths $\{\chi_{u,v}^l \mid u \to v \in A\}$, we aim to construct a schedule for locomotive l by finding a shortest path from vertex $\bar{o}$ to vertex $\bar{d}$ in network G subject to a side constraint that each train $k \in K$ can be picked up by l at most once. This side constraint is necessary because $\chi^l_{u,v}$ may be negative for some pickup arc $u \to v$ and thus a shortest path from $\bar{o}$ to $\bar{d}$ without this side constraint may pick up a train more than once. We develop a labeling algorithm for finding such constrained shortest path for locomotive l. Labeling algorithms have been widely used in solving constrained shortest path problems in vehicle routing analyses (see, e.g., Ioachim et al. 1998 and Righini and Salani 2008). For each $u \in V$, we introduce a two-dimensional label (K_u, F_u) to indicate the status of a partial path from vertex $\bar{o}$ to vertex u with no train being picked up more than once, where K_u indicates the set of trains that have been picked up, and F_u indicates the length of the partial path. Vertex $\bar{o}$ has only one label, while each of the other vertices may have multiple labels. Initially, we set the label of vertex $\bar{o}$ to $(K_{\bar{o}}, F_{\bar{o}}) = (\emptyset, 0)$. For any label (K_u, F_u) of $u \in V$ that has been obtained, if arc $u \to v \in A$ is not a pickup arc of any train $k \in K_u$, then we can extend the labeling to vertex $v \in V$ with the label for vertex v being (K_v, F_v), where

$$K_v = \begin{cases} K_u \cup \{k\}, & \text{if } u \to v \text{ is a pickup arc of train } k \in K \setminus K_u; \\ K_u, & \text{otherwise;} \end{cases}$$

and

$$F_v = F_u + \chi^l_{u,v}.$$

By repeatedly extending the labels, we can generate labels for all vertices in V.

Noting that G is an acyclic network, we extend the labeling of the vertices following a topological vertex order of G from $v = \bar{o}$ to $v = \bar{d}$. In order to save computational time, we sacrifice the optimality and find only an approximated constrained shortest path for locomotive l. This is achieved by imposing a restriction on the labeling algorithm, so that for each vertex $v \in V$, only the label (K_v, F_v) with the smallest F_v value is maintained. We refer to this algorithm as the *restricted labeling algorithm*.

The collection of the paths (or schedules) constructed in Phase 1 for all the locomotives forms a feasible solution to problem $\mathbf{P}$. In Phase 2, we aim to obtain a better feasible solution by identifying locomotives with schedules that can be improved, where the schedule of a locomotive $l \in L$ corresponds to a path in network G and the length of the path is measured using the arc lengths $\{\xi^l_{u,v} \mid u \to v \in A\}$. Similar to a local search algorithm, we consider each locomotive l in an arbitrary sequence and examine whether or not the schedule of locomotive l can be replaced by any improved schedule, while keeping the schedules of other locomotives unchanged. To ensure that the improved schedule of locomotive l does not violate any incompatibility constraint of problem $\mathbf{P}$, we need to prevent its corresponding path in network G from traversing arcs in an arc subset $\bar{C}^l_2$. Arc subset $\bar{C}^l_2$ has the same definition as $\bar{C}^l_1$, except that $\bar{C}^l_2$ is defined based on the schedules of all locomotives other than l, while $\bar{C}^l_1$ is defined based on the schedules of the locomotives with higher ranks than l. We revise the lengths of the arcs in $\bar{C}^l_2$ to $+\infty$ accordingly and let K^l be the set of trains assigned to locomotive l in the schedule obtained in Phase 1. In order to ensure that the improved schedule of locomotive l obtained in Phase 2 can serve at least all those trains that have already been assigned to it by Phase 1, we let the trains in K^l have higher priorities to be assigned to l by revising the lengths of pickup arcs of these trains to $-M$, where M is a large number. Thus, for arc $u \to v \in A$, the revised arc length $\chi^l_{u,v}$ is defined as follows:

$$\chi^l_{u,v} = \begin{cases} +\infty, & \text{if } u \to v \in \bar{C}^l_2; \\ -M, & \text{if } u \to v \text{ is a pickup arc of a train in the set } K^l; \\ \xi^l_{u,v}, & \text{otherwise.} \end{cases}$$

With the arc lengths $\{\chi^l_{u,v} \mid u \to v \in A\}$ defined above, we apply the restricted labeling algorithm described in Phase 1 to find an approximated constrained shortest path for locomotive l. This approximated constrained shortest path is a new schedule for locomotive l. If this new schedule has a length (measured using arc lengths $\xi^l_{u,v}$) smaller than that of the schedule obtained in Phase 1, then we assign this new schedule to locomotive l.

The heuristic returns a collection of the schedules for the locomotives in L, which forms a feasible solution to problem **P**, and the objective value of this feasible solution is an upper bound on the optimal objective value of **P**.

5 Computational Study

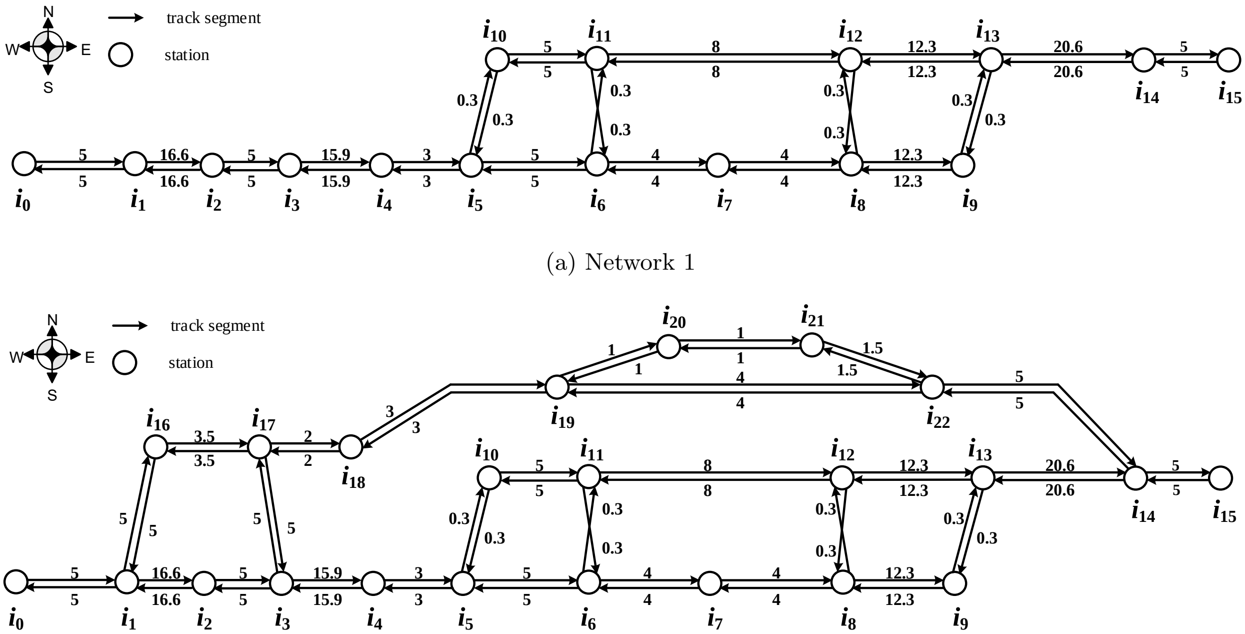
We implement our Lagrangian relaxation heuristic in C# using a personal computer with a 3.40 GHz quad-core processor (Intel Core i7-6700 Processor) and 32 GB RAM. We conduct a computational study to evaluate the performance of the heuristic. In addition to reporting the optimality gaps of the heuristic solutions, we compare the performance of the heuristic with two benchmark solution methods. The generation of test data is described in Section 5.1. Descriptions of the benchmark solution methods are provided in Section 5.2, and our computational results are presented in Section 5.3.

5.1 Generation of Test Instances

In our computational study, we adopt the two railway networks presented in Meng and Zhou (2014, Sec. 6.1), where the first network is originated from INFORMS RAS (2012), and the second network is an extension of the first one with a larger size. In order to convert these two single-track railway networks into our model's setting, we modify them as follows:

- Each siding track and its corresponding main track in the given single-track railway network are regarded as a station in our double-track railway network.
- Each joint point in the given single-track railway network is regarded as a station in our double-track railway network.
- All stations are uncapacitated.
- Each group of serial arcs between two stations is regarded as a track segment in our double-track railway network, and the length of this track segment is equal to the total length of the serial arcs in this group.
- All track segments are changed from single-track to double-track.

The two double-track railway networks obtained, namely Network 1 and Network 2, are shown in Figure 4. Network 1 has 16 stations $i_0, i_1, \ldots, i_{15}$ and 36 track segments, while Network 2 has 23 stations $i_0, i_1, \ldots, i_{22}$



(a) Network 1

(b) Network 2

Figure 4: Double-track networks modified from Meng and Zhou's (2014) single-track networks.

and 56 track segments. Each station i has a 4-minute minimum headway for arrivals (i.e., $g_i = 4$) and a 2-minute minimum headway for departures (i.e., $h_i = 2$). The length (in miles) of each track segment is shown on the corresponding arc in the two network diagrams. Note that some of the arcs in these two networks such as $i_6 \rightarrow i_{11}$ and $i_{11} \rightarrow i_6$ correspond to crossovers in Meng and Zhou's (2014) original networks, and thus they are relatively short. In reality, the two ends of a crossover are not two separate train stations. However, for simplicity, in Networks 1 and 2, we treat the two ends of each crossover as separate stations. For test instances with underlying Network 1, we set the length of the planning horizon to $T = 720$ minutes (i.e., 12 hours). For test instances with underlying Network 2, we set $T = 1440$ minutes (i.e., 24 hours).

For test instances with underlying Network 1 and Network 2, we consider the 4 train routes R_1, R_2, R_3, R_4 and the 8 train routes $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8$, respectively, as listed in Table 3. Routes R_1 and R_2 correspond to the two different directions (i.e., eastbound and westbound) of the longest path in these networks. Routes R_3 and R_4 correspond to the two different directions of a shorter path that passes through stations i_{10}, i_{11}, and

Table 3: Data for computational study: Train routes.

R_1: $i_0 \rightarrow i_1 \rightarrow i_2 \rightarrow i_3 \rightarrow i_4 \rightarrow i_5 \rightarrow i_6 \rightarrow i_7 \rightarrow i_8 \rightarrow i_9 \rightarrow i_{13} \rightarrow i_{14} \rightarrow i_{15}$
R_2: $i_{15} \rightarrow i_{14} \rightarrow i_{13} \rightarrow i_9 \rightarrow i_8 \rightarrow i_7 \rightarrow i_6 \rightarrow i_5 \rightarrow i_4 \rightarrow i_3 \rightarrow i_2 \rightarrow i_1 \rightarrow i_0$
R_3: $i_5 \rightarrow i_{10} \rightarrow i_{11} \rightarrow i_{12} \rightarrow i_{13}$
R_4: $i_{13} \rightarrow i_{12} \rightarrow i_{11} \rightarrow i_{10} \rightarrow i_5$
R_5: $i_{16} \rightarrow i_{17} \rightarrow i_{18} \rightarrow i_{19} \rightarrow i_{22} \rightarrow i_{14}$
R_6: $i_{14} \rightarrow i_{22} \rightarrow i_{19} \rightarrow i_{18} \rightarrow i_{17} \rightarrow i_{16}$
R_7: $i_{18} \rightarrow i_{19} \rightarrow i_{20} \rightarrow i_{21} \rightarrow i_{22}$
R_8: $i_{22} \rightarrow i_{21} \rightarrow i_{20} \rightarrow i_{19} \rightarrow i_{18}$

Table 4: Data for computational study: Train types.

Train type	Speed multiplier	π_k	$\phi_k(\nu_k)$	γ_k	$p'_k - p_k$
Type 1	1	400	$2.5\nu_k$	5	15
Type 2	0.9	360	$2.0\nu_k$	4	15
Type 3	0.8	320	$1.5\nu_k$	3	15
Type 4	0.7	280	ν_k	2	20
Type 5	0.6	280	ν_k	2	20
Type 6	0.5	280	ν_k	2	20

i_{12}. Routes R_5 and R_6 correspond to the two different directions of a path in the top part of Network 2. Routes R_7 and R_8 correspond to the two different directions of a shorter path that passes through stations i_{20} and i_{21} of Network 2. For each test instance that belongs to Network 1, the route of train k is randomly selected from $\{R_1, \ldots, R_4\}$, where each route is selected with probability 1/4. For each test instance that belongs to Network 2, the route of train k is randomly selected from $\{R_1, \ldots, R_8\}$, where each route is selected with probability 1/8.

We set the maximum speed of the fastest train to 80 miles per hour (mph) and round the travel time of each train along a track segment to the nearest minute. We consider 6 train types as shown in Table 4, where each train type has a speed multiplier (see, e.g., INFORMS RAS 2012). For example, if train k's speed multiplier is 0.8, then train k's maximum speed is $(0.8)(80) = 64$ mph, and this train's minimum travel time required to traverse track segment $i_3 \to i_4$ with a length of 15.9 miles is $\alpha^k_{i_3 i_4} = (15.9/64) \cdot 60 \approx 15$ minutes. The minimum travel time required to traverse the short track segments $i_5 \to i_{10}$, $i_{10} \to i_5$, $i_6 \to i_{11}$, $i_{11} \to i_6$, $i_8 \to i_{12}$, $i_{12} \to i_8$, $i_9 \to i_{13}$, and $i_{13} \to i_9$ is set equal to 1 minute. For each train k in a test instance, we randomly select one of these 6 train types, where the probability of each type being selected is 1/6. The cancellation penalty π_k, the shift penalty function $\phi_k(\cdot)$, the unit stretch penalty γ_k, and the departure time window size $p'_k - p_k$ of each train type are provided in Table 4. The earliest allowed departure time p_k of train k at the origin station o_k is randomly generated from a discrete uniform distribution between 0 and $T - 240$. The arrival time window $[q_k, q'_k]$ of train k at the destination station d_k is set equal to $[0, T]$; that is, there is no time window constraint imposed on the arrival time. The minimum required dwell time β_{ki} of train k at each station i is randomly selected from the set $\{0, 3, 6\}$, where the probability of each value being selected is 1/3. The ideal timetable of train k is the timetable that departs from o_k at time p_k and traverses the train path with minimum duration.

We set the maximum speed of the fastest locomotive to 80 mph. We consider 2 locomotive types with speed multiplier 1 and 0.7 respectively. Locomotives with speed multiplier 1 can serve trains of all types, while locomotives with speed multiplier 0.7 can serve trains of types 4, 5, and 6. For each locomotive l in a test instance, we randomly select one of these 2 locomotive types, where the probability of each type being selected is 1/2. Then, the values of parameters Ω_k and $\hat{\alpha}^l_{ij}$ can be obtained accordingly. The origin-destination pair of locomotive l is randomly selected from the set $\{(i_0, i_0), (i_{15}, i_{15}), (i_0, i_{15}), (i_{15}, i_0)\}$, where the probability of each pair being selected is 1/4. Other parameters of locomotive l are set equal to the following: $\hat{p}_l = 0$, $\hat{q}_l = T$, $c_l = 1$, $c'_l = 0.9$, $f_{lk} = 20$ (for each train k), and $\epsilon^+_{lk} = \epsilon^-_{lk} = 8$ (for each train k).

In our computational study, we consider two railway networks (i.e., Networks 1 and 2) and seven problem sizes with $|K| = 16, 32, 48, 64, 80, 96, 112$. Thus, there are 14 combinations of railway network and problem size. For Network 1, we set $|L| = 6, 12, 18, 24, 30, 36, 42$ for the seven problem sizes, respectively. Since the train routes in Network 2 are shorter than the train routes in Network 1 on average, we use a smaller number of locomotives for Network 2. Thus, for Network 2, we set $|L| = 3, 6, 9, 12, 15, 18, 21$ for the seven problem sizes, respectively. For each combination, we generate 5 random test instances. Hence, there are 70 test instances in total. We solve each test instance using the abovementioned Lagrangian relaxation heuristic and two benchmark solution methods described in Section 5.2. For the solutions of each test instance obtained by these methods, and we determine their optimality gaps defined as

$$Gap = \frac{UB - LB}{LB} \times 100\%,$$

where LB is the lower bound value generated by the Lagrangian relaxation procedure, and UB is the objective value of the corresponding solution method. We also record the number of canceled trains, the average shift of the uncanceled trains, and the average stretch of the uncanceled trains, denoted as $|K_c|$, $\bar{\nu}$, and $\bar{\mu}$, respectively, of each heuristic solution.

In our implementation of the subgradient optimization procedure, parameter a is set to 1.5. The initial value of the step size parameter θ is set to 2. In the solution process, θ is reduced by 20% if the best lower bound identified has no improvement for 10 consecutive iterations. The computation is terminated if it meets one of the following conditions: (1) the number of iterations has reached 1000, or there is no improvement of the best lower bound for 100 consecutive iterations; (2) the best upper bound identified does not exceed 101% of the best lower bound identified by the subgradient optimization procedure. For the first 300 iterations of the subgradient optimization procedure, we execute the upper bound heuristic and update UB at each iteration. After 300 iterations, since the upper bound heuristic becomes less likely to be able to identify a better upper bound, we execute the upper bound heuristic only with probability 0.1 at each iteration.

5.2 Benchmark Solution Methods

In the computational study, we compare the performance of our solution method with that of two benchmark methods. The first benchmark method is a priority heuristic (or PR for short), which constructs schedules for locomotives one by one based on a pre-determined priority sequence of locomotives. This heuristic is simple and efficient. It mimics the manual decision making process where locomotives are scheduled one after another. To implement this heuristic, we first relax the incompatibility constraints (5) of problem $\mathbf{P}$. The relaxed problem, which is equivalent to the Lagrangian relaxation problem $\mathbf{P}(\lambda)$ with $\lambda_C = 0$ for $C \in \mathcal{C}$, can be decomposed into $|L|$ shortest path problems on the acyclic network G. The decomposed problem corresponding to locomotive l, denoted $\mathbf{P}_l(\mathbf{0})$, has arc lengths $\{\xi^l_{u,v} \mid u \to v \in A\}$. A priority sequence of the locomotives is then determined based on the nondecreasing order of the shortest path lengths of problems $\{\mathbf{P}_l(\mathbf{0}) \mid l \in L\}$, with ties broken arbitrarily. That is, locomotives corresponding to smaller shortest path lengths are of higher priority. Following this sequence, heuristic PR applies the restricted labeling algorithm described in Section 4.3 to construct schedules for the locomotives one by one. When the heuristic constructs the schedules for locomotives with lower priorities, the schedules for locomotives with higher priorities are kept unchanged.

The second benchmark method is a sequential heuristic (or SQ for short) that solves the problem by following a traditional approach, where train timetabling decision is made before the locomotive assignment decision. This heuristic first solves a train timetabling subproblem, followed by a locomotive assignment subproblem. Since both subproblems are special cases of problem **P**, we can apply the Lagrangian relaxation heuristic described in Section 4 to each subproblem. More specifically, since the train timetabling subproblem is solved without taking into account the locomotives' availability, this subproblem is equivalent to the special case of problem **P** in which each train $k \in K$ can only be served by a distinct dummy locomotive l_k. Thus, in this subproblem, we set the compatible locomotive set to $\Omega_k = \{l_k\}$ for each train k. For each dummy locomotive l_k, we set the origin station $\hat{o}_{l_k}$ to o_k and the destination station $\hat{d}_{l_k}$ to d_k. We also set the earliest start time of operation to $\hat{p}_{l_k} = 0$, the latest completion time of operation to $\hat{q}_{l_k} = T$, the minimum time to traverse each arc $i \to j$ when light-running to $\hat{\alpha}_{ij}^{l_k} = 0$, the fixed pickup and drop-off service times to $\epsilon_{l_k k}^{+} = \epsilon_{l_k k}^{-} = 0$, the fixed assignment cost to $f_{l_k k} = 0$, and the operating costs per unit time to $c_{l_k} = c'_{l_k} = 0$. A solution of the train timetabling subproblem can be obtained by solving this revised problem **P** with the locomotive set $\{l_1, \ldots, l_{|K|}\}$ via the Lagrangian relaxation heuristic described in Section 4. We let $K' \subseteq K$ denote the subset of trains in this solution that have feasible timetables.

After obtaining the timetable for each train $k \in K'$ and canceling the train in $K \setminus K'$, we need to assign locomotives to the trains in K'. This locomotive assignment subproblem is equivalent to a special case of problem **P**, in which each train $k \in K'$ is either canceled or assigned a locomotive in L. When a train k is assigned a locomotive in L, it needs to follow its timetable generated by the train timetabling subproblem. This is achieved by revising the arc lengths in G to $+\infty$ for those loaded-running arcs $(k, \rho'(i), t) \to (k, \rho(j), t')$, dwelling arcs $(k, \rho(i), t) \to (k, \rho'(i), t')$, and dwelling arcs $(k, \rho'(i), t) \to (k, \rho'(i), t+1)$ that are not included in the timetable of train k, for each $k \in K'$. Then, the locomotive assignment subproblem can be solved by solving problem **P** with these revised arc lengths. Thus, we apply the Lagrangian relaxation heuristic described in Section 4 to determine the locomotive assignment and the locomotive schedule.

In our implementation of Heuristic SQ, the maximum number of iterations for solving the train timetabling subproblem and that for solving the locomotive assignment subproblem are both set to 500, so that the maximum total number of iterations is 1000, the same as the maximum number of iterations we set for the Lagrangian relaxation heuristic when solving the integrated problem **P**. In the subgradient optimization procedure of both subproblems, parameter a is set to 1.5. The initial value of the step size parameter θ is set to 2. In the solution process, θ is reduced by 20% if the best lower bound identified has no improvement for 10 consecutive iterations. For the first 150 iterations, we execute the upper bound heuristic and update UB at each iteration. After 150 iterations, we execute the upper bound heuristic only with probability 0.1 at each iteration.

5.3 Computational Results

Table 5 summarizes the computational results with each row representing the mean result of 5 test instances. The "*Gap*" column reports the mean optimality gap. The "$|K_c|$" column reports the mean number of canceled trains. The "$\bar{\nu}$" column reports the mean value of $\bar{\nu}$, where $\bar{\nu}$ is the average shift of the uncanceled trains in a test instance. The "$\bar{\mu}$" column reports the mean value of $\bar{\mu}$, where $\bar{\mu}$ is the average stretch of the uncanceled trains in a test instance. The "κ" column reports the mean value of

$$\kappa = \frac{\sum_{l \in L} (\text{amount of serving time of locomotive } l)}{\sum_{l \in L} (\text{amount of available time of locomotive } l)} \times 100\%,$$

where the available time of locomotive l equals $\hat{q}_l - \hat{p}_l$, and the serving time of locomotive l includes the amount of time locomotive l spends on loaded-running, light-running, dwelling, picking up trains, and dropping off trains. Thus, κ measures the percentage of available time the locomotives are utilized. The "Time" column reports the mean computational time (in CPU seconds) for solving a test instance in the problem set.

From these computational results, we observe that Heuristic PR is highly efficient, but the optimality gaps of the solutions that it generates are significantly larger than those generated by the Lagrangian relaxation heuristic. The solutions generated by Heuristic PR have higher $\bar{\nu}$ and $\bar{\mu}$ values than the Lagrangian relaxation heuristic solutions, implying that Heuristic PR generates solutions with worse train schedules. We also observe that the solutions generated by Heuristic PR tend to have higher $|K_c|$ values and lower κ values than the Lagrangian relaxation heuristic solutions, implying that Heuristic PR tends to generate solutions with more canceled trains and lower locomotive utilization. Both the Lagrangian relaxation heuristic and Heuristic SQ require a lot of computational time, but the solutions generated by Heuristic SQ have significantly larger optimality gaps. This observation suggests that making train timetabling and locomotive assignment decisions

Table 5: Computational results.

		Lagrangian relaxation heuristic						Heuristic PR						Heuristic SQ					
$\|L\|$	$\|K\|$	*Gap*	$\|K_c\|$	$\bar{\nu}$	$\bar{\mu}$	κ	Time	*Gap*	$\|K_c\|$	$\bar{\nu}$	$\bar{\mu}$	κ	Time	*Gap*	$\|K_c\|$	$\bar{\nu}$	$\bar{\mu}$	κ	Time
Network 1																			
6	16	0.8%	3.2	1.98	0.00	38%	19.4	8.3%	4.2	2.30	0.19	34%	0.2	5.8%	4.0	0.16	0.04	36%	82.4
12	32	1.8%	4.0	3.37	0.41	41%	220.4	10.3%	6.4	3.81	0.53	36%	0.6	10.2%	7.0	0.61	0.28	37%	339.8
18	48	4.4%	8.0	3.37	0.44	36%	623.8	12.3%	9.6	4.09	0.61	36%	1.2	12.7%	11.6	1.21	0.66	34%	1023.8
24	64	8.5%	5.6	4.12	0.84	41%	1313.8	17.4%	8.4	4.07	1.46	40%	2.4	16.1%	11.2	1.96	0.68	36%	1538.3
30	80	10.9%	9.8	4.62	1.20	39%	2185.0	17.3%	13.0	5.25	1.63	38%	3.4	20.8%	18.8	2.38	0.94	34%	3200.6
36	96	15.6%	10.8	5.30	1.44	38%	3702.4	22.6%	11.0	5.90	2.84	40%	5.4	26.6%	20.4	3.05	1.26	33%	4041.4
42	112	16.8%	15.6	5.28	1.64	38%	5733.8	22.9%	19.2	5.99	2.45	37%	7.7	30.8%	29.4	2.97	1.62	32%	7093.2
Network 2																			
3	16	0.7%	3.2	1.33	0.00	25%	14.4	5.6%	3.6	1.91	0.00	24%	0.2	11.4%	4.8	0.14	0.00	20%	70.9
6	32	0.9%	3.8	1.54	0.04	28%	159.2	4.5%	3.6	1.91	0.03	29%	0.5	9.8%	5.4	0.07	0.01	27%	312.6
9	48	1.4%	3.4	1.93	0.03	30%	641.8	8.2%	3.4	1.44	0.02	31%	1.0	9.0%	5.8	0.13	0.00	28%	1492.9
12	64	2.0%	4.2	2.68	0.05	30%	844.2	12.0%	5.4	2.55	0.12	30%	1.7	12.5%	7.6	0.27	0.03	28%	3340.9
15	80	4.9%	1.2	2.74	0.11	32%	1608.2	13.6%	2.0	3.15	0.22	32%	2.8	20.2%	6.4	0.28	0.11	29%	3864.5
18	96	5.9%	3.0	2.79	0.10	32%	2497.4	14.9%	3.8	3.40	0.15	32%	3.8	19.4%	9.8	0.47	0.05	29%	6604.1
21	112	9.8%	2.6	3.32	0.32	31%	3790.0	15.2%	2.4	4.00	0.36	32%	5.9	23.4%	9.2	0.65	0.16	29%	10090.7

Table 6: Lagrangian relaxation heuristic solutions under different values of $|L|$.

Network 1								Network 2							
$\|L\|$	$\|K\|$	*Gap*	$\|K_c\|$	$\bar{\nu}$	$\bar{\mu}$	κ	Time	$\|L\|$	$\|K\|$	*Gap*	$\|K_c\|$	$\bar{\nu}$	$\bar{\mu}$	κ	Time
16	64	5.1%	9.8	3.96	0.34	53%	750.0	8	64	1.7%	8.8	2.59	0.03	40%	481.2
20	64	6.3%	6.8	4.13	0.40	46%	983.6	10	64	2.1%	6.0	2.58	0.03	34%	659.8
24	64	8.1%	4.0	4.30	0.81	41%	1201.8	12	64	2.2%	4.6	2.40	0.05	30%	917.2
28	64	8.8%	4.2	4.05	0.83	35%	1490.2	14	64	2.2%	2.2	2.66	0.08	27%	1117.0
32	64	9.4%	4.0	4.33	0.69	31%	1693.8	16	64	2.6%	1.2	2.42	0.04	24%	1337.4

separately will result in solutions with much worse quality. The solutions generated by Heuristic SQ have lower $\bar{\nu}$ and $\bar{\mu}$ values than the solutions generated by the Lagrangian relaxation heuristic but with a significantly higher number of canceled trains. This implies that making locomotive assignment decision after making train timetabling decision could result in more train cancellations.

From Table 5, we also observe that the performance gap of all three heuristics tend to be larger as $|L|$ and $|K|$ increase; that is, as traffic in the railway network increases. This indicates that either the problem becomes harder to solve as traffic increases, or the Lagrangian relaxation lower bound is less tight as traffic increases.

Next, we study how the computational results are affected by the number of available locomotives. We consider the case with $|K| = 64$ and rerun the experiments by varying the value of $|L|$, with the other parameters kept unchanged. Table 6 summarizes the results with each row representing the mean result of 5 test instances. From these results, we observe that both the number of canceled trains and the locomotive utilization tend to decrease as the number of locomotive increases, but the $\bar{\nu}$ and $\bar{\mu}$ indicate that the quality of the train schedules is not necessarily improved by having more locomotives. We also observe that both the optimality gaps of the Lagrangian relaxation solutions and the computational time of the Lagrangian relaxation heuristic tend to increase as the number of locomotives increases. This indicates that both the effectiveness and efficiency of the heuristic can be affected by an increase in number of locomotives. Nevertheless, the results in Tables 5 and 6 demonstrate that the Lagrangian relaxation can generate fairly effective solutions to all the test instances. From these results, we believe that using a more powerful computing facility, and allowing the heuristic to run for more iterations, better solutions and lower bounds can be obtained for larger-size instances.

6 Extensions

In this section, we discuss several extensions of our model and their solution methods.

6.1 Generating Solutions with Cyclic Locomotive Schedules

In some railway applications, the train timetable is repeated daily or weekly for a certain period of time. In such applications, it is more desirable to have each locomotive finish a cycle's operation at a station at which the locomotive can start its next cycle's operation. To satisfy this requirement, one approach is to simply set $\hat{d}_l = \hat{o}_l$ for each locomotive l. A drawback of this approach is that it allows no flexibility in selecting the destination stations for the locomotives. An alternative approach is as follows. For each group of identical locomotives

$\{l_1, l_2, \ldots, l_\zeta\}$, the locomotives' origin stations $\hat{o}_{l_1}, \hat{o}_{l_2}, \ldots, \hat{o}_{l_\zeta}$ are given, but the decision maker may choose these locomotives' destination stations $\hat{d}_{l_1}, \hat{d}_{l_2}, \ldots, \hat{d}_{l_\zeta}$, subject to the constraint that $(\hat{d}_{l_1}, \hat{d}_{l_2}, \ldots, \hat{d}_{l_\zeta})$ must be a permutation of $(\hat{o}_{l_1}, \hat{o}_{l_2}, \ldots, \hat{o}_{l_\zeta})$.

A given problem with this new setting can be transformed into our model, and is therefore solvable by our Lagrangian relaxation heuristic, as follows. Consider any instance of this new problem. For ease of presentation, we assume, without loss of generality, that $\hat{q}_l < T$ for each locomotive $l \in L$. For each group of identical locomotives $\{l_1, l_2, \ldots, l_\zeta\}$, we add a dummy station d' and add a dummy track segment $\hat{o}_{l_i} \rightarrow d'$ for each $i = 1, 2, \ldots, \zeta$ to the railway network N. For $i = 1, 2, \ldots, r$, the time for locomotives $l_1, l_2, \ldots, l_\zeta$ to traverse the dummy track segment $\hat{o}_{l_i} \rightarrow d'$ when light-running is set equal to $T - \hat{q}_{l_i}$, while the time for locomotives in $L \setminus \{l_1, l_2, \ldots, l_\zeta\}$ to traverse $\hat{o}_{l_i} \rightarrow d'$ when light-running is $+\infty$. The minimum headway between arrivals from the same track segment at station d' is set equal to T. The destination stations of locomotives $l_1, l_2, \ldots, l_\zeta$ are changed to d', and the latest completion time of operation of locomotives $l_1, l_2, \ldots, l_\zeta$ are changed to T.

In the transformed problem, for $i = 1, 2, \ldots, r$, because the latest completion time of operation of locomotive l_i is T and the time for l_i to traverse a dummy track segment is $T - \hat{q}_{l_i}$, locomotive l_i needs to complete its service at one of stations $\hat{o}_{l_1}, \hat{o}_{l_2}, \ldots, \hat{o}_{l_\zeta}$ no later than $\hat{q}_{l_i}$. Since the minimum headway between arrivals from the same track segment at station d' is T, at most one locomotive can traverse each dummy track segment $\hat{o}_{l_j} \rightarrow d'$ for $j = 1, 2, \ldots, r$. Hence, a feasible solution of the original problem with a total cost Γ exists if and only if a feasible solution of the transformed problem with a total cost $\Gamma + \sum_{l \in L} c_l(T - \hat{q}_l)$ exists, where the constant term "$\sum_{l \in L} c_l(T - \hat{q}_l)$" is the total operating cost for the locomotives to run on the dummy track segments. Therefore, solving the transformed problem yields an optimal solution of the original problem.

6.2 Handling Train Services that Require Multiple Locomotives

In some railway applications, there are situations in which a train service requires more than one locomotive, where grouping and splitting of locomotives are required. To model this requirement, consider the situation where each element ω of Ω_k is a subset of locomotives that can put together to serve train k simultaneously. For example, if $\omega = \{l_1, l_2, \ldots, l_\zeta\} \in \Omega_k$, then locomotives $l_1, l_2, \ldots, l_\zeta$ can form a consist for serving train k. For any $k \in K$ and any $\omega \in \Omega_k$, let $c_{\omega k}$ be the operating cost per unit time when consist ω is running with train k, let $c'_{\omega k}$ be the operating cost per unit time when consist ω is serving train k but is not running, let $f_{\omega k}$ be the fixed cost for assigning consist ω to train k, let $\epsilon^+_{\omega k}$ be the amount of time needed for assembling consist ω and picking up train k, and let $\epsilon^-_{\omega k}$ be the amount of time need for dropping off train k and disassembling consist ω. We let $\hat{p}_{\omega k} = \max_{l \in \omega} \hat{p}_l$ and $\hat{q}_{\omega k} = \min_{l \in \omega} \hat{q}_l$ denote the earliest start time and the latest completion time, respectively, of consist ω's operation. We let $\hat{\alpha}^{\omega k}_{ij}$ denote the running time for consist ω to traverse track segment $i \rightarrow j$ with train k, for $i \rightarrow j \in E$.

To represent this extended problem by a state-space-time network, we modify network $G = (V, A)$ in such a way that a path from $\bar{o}$ to $\bar{d}$ may either represent a sequence of changes in state and location of a locomotive over time, or represent a sequence of changes in state and location of a consist over time. To do so, each arc

$u \to v \in A$ has not only a cost coefficient $\xi^l_{u,v}$ for each locomotive l but also a cost coefficient $\xi^{\omega k}_{u,v}$ for each train $k \in K$ and each consist $\omega \in \Omega_k$. Different types of arcs in G and their cost coefficients are modified as follows:

- Starting and ending arcs: We construct the starting arcs that connect $\bar{o}$ to the locomotives' origin stations and the ending arcs that connect the locomotives' destination stations to $\bar{d}$ the same way as described in Section 3.1. In addition to these arcs, we add a new set of starting and ending arcs for the consists. For each $i \in S$ and $t = 0, 1, \ldots, T$, there is a new starting arc $\bar{o} \to (0, \rho(i), t)$ if this arc is not yet in network G and if there exists at least one train $k \in K$ and one consist $\omega \in \Omega_k$ such that $i = o_k$ and $t \geq \hat{p}_{\omega k}$. For each $k \in K$ and $\omega \in \Omega_k$, if $i = o_k$ and $t \geq \hat{p}_{\omega k}$, then $\xi^{\omega k}_{\bar{o},(0,\rho(i),t)} = 0$; otherwise, $\xi^{\omega k}_{\bar{o},(0,\rho(i),t)} = +\infty$. This arc allows consist ω to start its operation for serving train k at station o_k at or after time $\hat{p}_{\omega k}$. In each original starting arc $\bar{o} \to (0, \rho(i), t)$, if $\xi^{\omega k}_{\bar{o},(0,\rho(i),t)}$ is not yet defined for some k and ω, then we set $\xi^{\omega k}_{\bar{o},(0,\rho(i),t)}$ to $+\infty$. In each new starting arc $\bar{o} \to (0, \rho(i), t)$, if $\xi^{l}_{\bar{o},(0,\rho(i),t)}$ is not yet defined for some l, then we set $\xi^{l}_{\bar{o},(0,\rho(i),t)}$ to $+\infty$. New ending arcs are added to G in a similar fashion.
- Light-running and waiting arcs: We construct the light-running and waiting arcs the same way as described in Section 3.1. On these arcs, the cost coefficients $\xi^{\omega k}_{u,v}$ are set to $+\infty$, since consists are in operation only when serving trains.
- Pickup, drop-off, loaded-running, and dwelling arcs: The pickup, drop-off, loaded-running, and dwelling arcs are constructed by the same method as described in Section 3.1 but with the following differences. Instead of having these arcs constructed for each locomotive l using parameters $\hat{p}_l$, $\hat{q}_l$, $\max\{\hat{\alpha}^l_{ij}, \alpha^k_{ij}\}$, c_l, c'_l, f_{lk}, ϵ^+_{lk}, and ϵ^-_{lk}, these arcs are now constructed for each consist ω using parameters $\hat{p}_{\omega k}$, $\hat{q}_{\omega k}$, $\hat{\alpha}^{\omega k}_{ij}$, $c_{\omega k}$, $c'_{\omega k}$, $f_{\omega k}$, $\epsilon^+_{\omega k}$, and $\epsilon^-_{\omega k}$, respectively. On these arcs, the cost coefficients $\xi^l_{u,v}$ are set to 0.

Let A_1 denote the set of all light-running and waiting arcs in network G, and A_2 denote the set of all pickup, drop-off, loaded-running, and dwelling arcs in network G. Set $\mathcal{C}$ is defined the same way as in Section 3.1, but the incompatibility constraints (5) are rewritten as follows:

$$\sum_{u\to v\in C\cap A_1} \sum_{l\in L} x^l_{u,v} + \sum_{u\to v\in C\cap A_2} \sum_{k\in K} \sum_{\omega\in\Omega_k} x^{\omega k}_{u,v} \leq 1, \quad \text{for all } C \in \mathcal{C}. \tag{7}$$

In other words, the incompatibility constraints on the light-running and waiting arcs are imposed on the $x^l_{u,v}$ variables, while the incompatibility constraints on the pickup, drop-off, loaded-running, and dwelling arcs are imposed on the $x^{\omega k}_{u,v}$ variables. The integer programming formulation $\mathbf{P}$ becomes:

$$\mathbf{P}': \text{ Minimize } \sum_{k\in K} \pi_k + \sum_{l\in L} \sum_{u\to v\in A} \xi^l_{u,v} x^l_{u,v} + \sum_{k\in K} \sum_{\omega\in\Omega_k} \sum_{u\to v\in A} \xi^{\omega k}_{u,v} x^{\omega k}_{u,v} \tag{8}$$

subject to (2), (3), (4), (6), and (7)

$$\sum_{\omega\in\Omega_k} \sum_{\{v:\bar{o}\to v\in A\}} x^{\omega k}_{\bar{o},v} \leq 1, \quad \text{for all } k \in K \tag{9}$$

$$\sum_{\{u:u\to v\in A\}} x^{\omega k}_{u,v} = \sum_{\{w:v\to w\in A\}} x^{\omega k}_{v,w}, \quad \text{for all } k \in K;\ \omega \in \Omega_k;\ v \in V \setminus \{\bar{o}, \bar{d}\} \tag{10}$$

$$x^l_{u,v} = \sum_{k\in K} \sum_{\{\omega\in\Omega_k: l\in\omega\}} x^{\omega k}_{u,v}, \quad \text{for all } l \in L;\ u \to v \in A_2. \tag{11}$$

$$x^{\omega k}_{u,v} \in \{0, 1\}, \quad \text{for all } k \in K;\ \omega \in \Omega_k;\ u \to v \in A \tag{12}$$

Constraints (9) ensure that at most one consist will serve each train. Constraints (10) are the flow balance constraints for the consists. Constraints (11) ensure that locomotive l traverses an arc in A_2 if and only if exactly one consist containing l is traversing that arc. Constraints (11), together with constraints (2), (3), (4), and (6), imply that a consist can serve at most one train at a time.

To extend our Lagrangian relaxation heuristic to problem $\mathbf{P}'$, besides relaxing the incompatibility constraints (7), we also relax constraints (11) to ensure that the relaxed problem is computationally tractable. Thus, there are $|\mathcal{C}| + |L| \cdot |A_2|$ Lagrangian multipliers in total. After relaxing these constraints, the Lagrangian relaxation problem can be decomposed into $|L| + |K|$ independent subproblems, each of them corresponds to either a locomotive $l \in L$ or a train $k \in K$. Each subproblem can be solved as a shortest path problem in the acyclic network G. We can apply the subgradient optimization procedure to obtain a lower bound on the optimal objective value of the extended model.

For any given vector of Lagrangian multipliers, after solving the relaxation problem, we obtain a path for each locomotive, as well as the timetable and consist assignment for each uncanceled train service. We may utilize this information to construct heuristic solutions of problem $\mathbf{P}'$. The construction of heuristic solutions can be challenging, since we need to satisfy not only the incompatibility constraints but also the new constraints (11). We therefore leave the detailed development of the heuristic as well as the performance test of the solution method to future study.

Note that in this extended model, we have assumed that consist disassembling is always needed when the consist finishes serve a train. However, in practice, consist busting can be avoided if the same consist can pick up another train at the same station after it drops off a train. In such a case, the state-space-time network and the incompatibility constraints need to be modified further.

6.3 Other Extensions

In our model we assume that all track segments consist of one-way tracks. However, in reality, some track segments may be bi-directional. Our model can be extended to handle bi-directional tracks as follows. We represent each bi-directional track segment between stations i and j by two arcs $i \to j$ and $j \to i$ in network N. Then, we impose incompatibility constraints on the corresponding light-running arcs and loaded-running arcs in the state-space-time network G to disallow $i \to j$ and $j \to i$ from being traversed simultaneously. This extension, however, will increase the size of the state-space-time network as well as the number of incompatibility constraints in problem $\mathbf{P}$, and will affect the performance of the Lagrangian relaxation heuristic.

In our model we also assume that the train stations are uncapacitated. In reality, the capacity of some stations may be limited. Our model can be extended to handle station capacity constraints as follows. Suppose station i can accommodate at most n_i locomotives or locomotive-train pairs at any time. Then, we add an incompatibility constraint to problem $\mathbf{P}$ to limit the total flow along the dwelling arcs, waiting arcs, pickup arcs, and drop-off arcs that involve station i at each time point t to be at most n_i. The proposed Lagrangian relaxation heuristic can be applied to this extended problem. This extension will also increase the number of incompatibility constraints.

7 Conclusions

We have modeled and solved an integrated train timetabling and locomotive assignment problem with a general underlying railway network. The problem is formulated as a minimum cost multi-commodity network flow model with incompatible arcs and integer flow restrictions in a three-dimensional state-space-time network. We presented a Lagrangian relaxation heuristic for this network flow model. We conducted a computational study to test the performance of the Lagrangian relaxation heuristic, and compared the Lagrangian relaxation heuristic solutions with the solutions generated by two benchmark solution methods. The computational results demonstrate the effectiveness of the Lagrangian relaxation heuristic and report the benefits obtained by integrating the train timetabling and locomotive assignment decisions.

The state-space-time network is an important tool in our solution process, as it allows our model to be modified easily to accommodate additional train timetabling and locomotive planning requirements. For example, if some locomotives are not allowed to traverse certain track segments during certain time intervals, then the state-space-time network can be modified easily to accommodate this requirement by setting the cost of some arcs to infinity. One drawback of using this three-dimensional network is that the network size is very large, making the Lagrangian relaxation heuristic very time-consuming to execute. Hence, one interesting future research topic would be to develop mathematical techniques to identify the arcs in this three-dimensional network that can be eliminated without affecting the optimal solution of the problem.

Besides the extended model with multiple locomotives per train discussed in Section 6.2, several other extensions of our work are worth investigating. First, the objective function of our model includes various costs of each locomotive and each train. However, it does not attempt to balance locomotives' service times, and thus in the solution some locomotives may have much longer work hours than other locomotives. Extending our model to include an additional cost term to control the imbalance in the locomotives' service times and developing solution methods for the extended model is an interesting research topic. Second, although our Lagrangian relaxation heuristic can generate good solutions and tight lower bounds, it requires a lot of computational time. Thus, this solution method has its limitation when the underlying railway network is large. It would be worthwhile to develop more efficient solution methods such as meta-heuristics for integrated train timetabling and locomotive assignment based on the state-space-time network model proposed in this paper. Finally, applying our modeling framework and solution techniques to other logistics planning problems would also be of future research interest.

Acknowledgments

The authors thank three anonymous referees for their helpful comments. This work was supported in part by the Hong Kong Research Grants Council under the Theme-based Research Scheme (T32-101/15-R). The first author was a postdoctoral fellow at The Hong Kong Polytechnic University while this work was being done. He was also supported in part by the National Natural Science Foundation of China (71701062).

References

Ahuja, R.K., Liu, J., Orlin, J.B., Sharma, D., Shughart, L.A., 2005. Solving real-life locomotive-scheduling problems. Transportation Science 39(4), 503–517.

Booler, J.M.P., 1980. The solution of a railway locomotive scheduling problem. Journal of the Operational Research Society 31(10), 943–948.

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

Bussieck, M.R., Winter, T., Zimmermann, U.T., 1997. Discrete optimization in public rail transport. Mathematical Programming 79(1–3), 415–444.

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

Cacchiani, V., Caprara, A., Toth, P., 2010a. Scheduling extra freight trains on railway networks. Transportation Research Part B 44(2), 215–231.

Cacchiani, V., Caprara, A., Toth, P., 2010b. Solving a real-world train-unit assignment problem. Mathematical Programming 124(1–2), 207–231.

Cadarso, L., Maróti, G., Marín, Á., 2015. Smooth and controlled recovery planning of disruptions in rapid transit networks. IEEE Transactions on Intelligent Transportation Systems 16(4), 2192–2202.

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

Camerini, P.M., Fratta, L., Maffioli, F., 1975. On improving relaxation methods by modified gradient techniques. In: Balinski, M.L., Wolfe, P. (Eds.), Nondifferentiable Optimization (Mathematical Programming Study, Volume 3), North-Holland, Amsterdam, pp. 26–34.

Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and solving the train timetabling problem. Operations Research 50(5), 851–861.

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

Cordeau, J.-F., Desaulniers, G., Lingaya, N., Soumis, F., Desrosiers, J., 2001a. Simultaneous locomotive and car assignment at VIA Rail Canada. Transportation Research Part B 35(8), 767–787.

Cordeau, J.-F., Soumis, F., Desrosiers, J., 2001b. Simultaneous assignment of locomotives and cars to passenger trains. Operations Research 49(4), 531–548.

Cordeau, J.-F., Soumis, F., Desrosiers, J., 2000. A Benders decomposition approach for the locomotive and car assignment problem. Transportation Science 34(2), 133–149.

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

Dauzère-Pérès, S., De Almeida, D., Guyon, O., Benhizia, F., 2015. A Lagrangian heuristic framework for a real-life integrated planning problem of railway transportation resources. Transportation Research Part B 74, 138–150.

Florian, M., Bushell, G., Ferland, J., Guérin, G., Nastansky, L., 1976. The engine scheduling problem in a railway network. INFOR 14(2), 121–138.

Forbes, M.A., Holt, J.N., Watts, A.M., 1991. Exact solution of locomotive scheduling problems. Journal of the Operational Research Society 42(10), 825–831.

Ghoseiri, K., Ghannadpour, S.F., 2010. A hybrid genetic algorithm for multi-depot homogenous locomotive assignment with time windows. Applied Soft Computing 10(1), 53–65.

Godwin, T., Gopalan, R., Narendran, T.T., 2006. Locomotive assignment and freight train scheduling using genetic algorithms. International Transactions in Operational Research 13(4), 299–332.

Held, M., Karp, R.M., 1971. The traveling-salesman problem and minimum spanning trees: Part II. Mathematical Programming 1, 6–25.

INFORMS RAS, 2012. Problem Solving Competition: Movement Planner Algorithm Design for Dispatching on Multi-Track Territories. http://connect.informs.org/railway-applications/awards/problem-solving-competition/2012 (accessed August 20, 2018).

Ioachim, I., Gélinas, S., Soumis, F., Desrosiers, J., 1998. A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. Networks 31(3), 193–204.

Kroon, L., Huisman, D., Maróti, G., 2014. Optimisation models for railway timetabling. In: Hansen, I.A., Pachl, J. (Eds.), Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation, 2nd edition, DVV Media Group, Hamburg, Germany, pp. 155–173.

Kuo, C.-C., Nicholls, G.M., 2007. A mathematical modeling approach to improving locomotive utilization at a freight railroad. Omega 35(5), 472–485.

Lusby, R.M., Larsen, J., Ehrgott, M., Ryan, D., 2011. Railway track allocation: Models and methods. OR Spectrum 33(4), 843–883.

Mahmoudi, M., Zhou, X., 2016. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state-space-time network representations. Transportation Research Part B 89, 19–42.

Meng, L., 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 67, 208–234.

Nourbakhsh, S.M., Ouyang, Y., 2010. Optimal fueling strategies for locomotive fleets in railroad networks. Transportation Research Part B 44(8–9), 1104–1114.

Piu, F., Speranza, M.G., 2014. The locomotive assignment problem: A survey on optimization models. International Transactions in Operational Research 21(3), 327–352.

Raviv, T., Kaspi, M., 2012. The locomotive fleet fueling problem. Operations Research Letters 40(1), 39–45.

Righini, G., Salani, M., 2008. New dynamic programming algorithms for the resource constrained elementary shortest path problem. Networks 51(3), 155–170.

Szpigel, B., 1973. Optimal train scheduling on a single track railway. In: Ross, M. (Ed.), Operational Research '72, North-Holland, Amsterdam, pp. 343–352.

Vaidyanathan, B., Ahuja, R.K., Liu, J., Shughart, L.A., 2008a. Real-life locomotive planning: New formulations and computational results. Transportation Research Part B 42(2), 147–168.

Vaidyanathan, B., Ahuja, R.K., Orlin, J.B., 2008b. The locomotive routing problem. Transportation Science 42(4), 492–507.

Wright, M.B., 1989. Applying stochastic algorithms to a locomotive scheduling problem. Journal of the Operational Research Society 40(2), 187–192.

Xu, X., Li, K., Lu, X., 2017. Simultaneous locomotive assignment and train scheduling on a single-track railway line: A simulation-based optimization approach. Computers & Industrial Engineering, forthcoming, http://dx.doi.org/10.1016/j.cie.2017.11.002

Zhou, W., Teng, H., 2016. Simultaneous passenger train routing and timetabling using an efficient train-based Lagrangian relaxation decomposition. Transportation Research Part B 94, 409–439.

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

Zhu, E., Crainic, T.G., Gendreau, M., 2014. Scheduled service network design for freight rail transportation. Operations Research 62(2), 383–400.

Ziarati, K., Soumis, F., Desrosiers, J., Gélinas, S., Saintonge, A., 1997. Locomotive assignment with heterogeneous consists at CN North America. European Journal of Operational Research 97(2), 281–292.

Ziarati, K., Soumis, F., Desrosiers, J., Solomon, M.M., 1999. A branch-first, cut-second approach for locomotive assignment. Management Science 45(8), 1156–1168.