A Time-Space Formulation for the Locomotive Routing Problem at the Canadian National Railways

Pedro L. Miranda[a,*], Jean-François Cordeau[a], Emma Frejinger[b]

[a] *HEC Montréal and CIRRELT*
3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7 Canada
[b] *Université de Montréal and CIRRELT*
2920 chemin de la Tour, Montréal, H3T 1J4 Canada

Abstract

This paper addresses the locomotive routing problem, a large-scale railway optimization problem that aims to determine the optimal sequence of trains to be followed by each locomotive in a given fleet, while considering locomotive maintenance over a weekly planning horizon. By using commodity aggregation and flow decomposition techniques, we design a tractable integer linear program for the problem. The formulation is based on a time-space network representation of the problem that allows us to track the maintenance status of specific locomotives over the planning horizon and to manage locomotive assignments to trains based on their current maintenance status. It also considers locomotive repositioning, train connections, and utilization of third-party locomotives (i.e., foreign power). Computational experiments on real instances from the Canadian National Railways show that our model is tractable despite its size and can be solved optimally within reasonable computing times. Our methodology performs favorably when compared to historical data supplied by the industrial partner. The solutions satisfy train schedules and locomotive maintenance while requiring fewer locomotives and less repositioning.

Keywords: Locomotive scheduling, locomotive routing, railway transportation, network optimization, integer programming.

[*] Corresponding author

Email addresses: pedro.miranda@hec.ca (Pedro L. Miranda), jean-francois.cordeau@hec.ca (Jean-François Cordeau), emma.frejinger@cirrelt.ca (Emma Frejinger)

1. Introduction

Railroad transportation plays an important role in the economy, providing efficient and cost-effective freight services for the transportation of products and goods. It also offers numerous opportunities for employing optimization techniques to solve large, interrelated and complex problems at the different decision-making levels, such as, expanding the rail network, increasing line capacity, building or closing yards, scheduling locomotives, planning maintenance, dispatching trains, and managing crew (Ahuja et al., 2005a).

Among these problems, the locomotive scheduling problem stands out due to its crucial role for effective railroad transportation. The high cost of each locomotive and the large number of them make the locomotive fleet one of their most valuable assets, representing an investment in the order of billions of dollars for large railways. Consequently, developing and implementing effective optimization tools to support locomotive scheduling decisions is highly desirable.

In brief, the locomotive scheduling problem aims to assign a *consist* (i.e., a set of locomotives) to each train in a given schedule, providing sufficient power to pull it from its origin to its destination, while satisfying a variety of operational and business constraints at minimum cost (Ahuja et al., 2005b; Vaidyanathan et al., 2008a). Depending on the specific decisions to be made and on the length of the planning horizon, this problem can be considered at two main decision levels, namely, tactical and operational. Here, we will focus on the operational planning level, where locomotives must be assigned to trains while taking into account locomotive maintenance over the planning horizon, usually a week.

At the tactical level, where the problem is referred to as the locomotive assignment problem (LAP), locomotives are classified into types based on their main characteristics, such as horsepower, pulling capabilities, weight, number of axles, and cost, among others. The problem is to determine the number of locomotives of each type to assign to each train while taking into account, e.g., constraints on the fleet size for each locomotive type, power requirements for each train, compatibility between trains and locomotives types, and a balanced flow of locomotives through the network. Given that a typical train schedule is a weekly plan to be repeated over a three or four-month period, the ultimate goal of the LAP is to provide a guideline on how to assign locomotive types to trains and how to reposition them in the network so that the assignment is repeated every week.

Research on the LAP includes different modeling and algorithmic strategies. Cordeau et al. (2000, 2001) propose multicommodity flow-based models for simultaneous locomotive and passenger railcar assignment at Via Rail Canada, and solve them by applying Benders decomposition (Benders, 1962). For the problem faced by CSX Transportation, a major U.S. railroad, Ahuja et al. (2005b) propose an integer multicommodity flow-based formulation, where commodities correspond to locomotive types, and a heuristic methodology to find high-quality solutions. Vaidyanathan et al. (2008a) present an alternative formulation, where commodities represent locomotive consists instead of locomotive types. Piu et al. (2015) propose an optimization model to determine the set of consist types to include in consist-based formulations for the LAP. More recently, Ortiz-Astorquiza et al. (2019) propose a novel hybrid formulation that combines features from both locomotive-based and consist-based representations of the problem to solve the LAP at the Canadian National Railways (CN). Scheffler et al. (2020) propose a formulation that generalizes the locomotive- and consist-based models proposed by Ahuja et al. (2005b) and Vaidyanathan et al. (2008a), respectively, while satisfying practical requirements commonly found in European freight railroads. We refer to Piu and Speranza (2014) for a survey on the LAP.

In practice, the output of the LAP cannot be directly implemented as it does not consider locomotive maintenance, which might reduce the locomotive availability over the planning horizon. Moreover, in real environments, planners are concerned with assigning locomotive units (i.e., individual and uniquely identified locomotives) to trains, rather than locomotive types. Therefore, one needs to go one step further and solve the so-called Locomotive Routing Problem (LRP) that arises at the operational level. In this problem, one needs to decide the sequence of trains each specific locomotive should operate, while respecting the consist type assigned to each train, locomotive maintenance, and a balanced flow of locomotives through the network so as to operate a weekly train schedule at minimum cost. The LRP does not change the consist type assigned to each train, which is an output from the LAP. Rather, it determines which specific locomotives make up the consist. Furthermore, it reoptimizes the locomotive flows through the network by taking into account changes in fleet size and locomotive availability caused by scheduled maintenance activities.

Despite the importance of solving the LRP to obtain implementable locomotive schedules in practice, the literature on this subject is rather scarce. Ziarati et al. (1997) study the problem arising at CN. They propose a multicommodity flow-based model, and reformulate it using Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960). This reformulation is solved heuristically by a branch-and-bound procedure, where the linear programming relaxation at each node is solved by column generation. In a subsequent work, Ziarati et al. (1999) present a cutting plane methodology for the same problem, which yields lower integrality gaps and shorter computing times. Rouillon et al. (2006) also build upon the work of Ziarati et al. (1997), and propose alternative branching and search strategies to enhance their heuristic branch-and-bound. Computational experiments highlight the savings, in terms of number of locomotives, obtained by the enhanced algorithm.

Vaidyanathan et al. (2008b) study the LRP at CSX Corporation, and propose a methodology to determine locomotive paths, taking into account fueling and maintenance constraints. The procedure is based on the *a priori* generation of locomotive paths that are guaranteed to satisfy those constraints. The enumerated paths are then used as input for an integer linear program that decomposes the LAP assignment into flows on paths.

More recently, Powell et al. (2014) and Bouzaiene-Ayari et al. (2016) propose an approach based on Approximate Dynamic Programming (ADP) to solve the LRP at Norfolk Southern. Besides locomotive maintenance and foreign power, their methodology also handles uncertainty on transit times, train and yard delays, and locomotive failures. A drawback of this strategy, as pointed out by the authors, is that ADP, despite being suitable to handle high levels of details, does not globally optimize the locomotive flows on the network over time.

In this research, we propose a modeling framework for the deterministic LRP. It is a compact integer multicommodity flow-based model that optimizes locomotive flows over the entire network, while considering scheduled locomotive maintenance. Unlike other approaches proposed in the literature, our methodology does not resort to algorithmic strategies that generate maintenance-feasible paths in advance (Vaidyanathan et al., 2008b) or implementing tailor-made solution methods (Powell et al., 2014). Rather, it is a simple yet effective tool that planners can use to support decision-making and analyze different operational scenarios within a few minutes. It makes use of commodity aggregation and flow decomposition techniques in order to devise a tractable integer programming formulation, which provides optimal solutions to real-size problems within short computing times when solved by state-of-the-art commercial solvers. This paper focuses on the LRP faced by CN, a major North American railroad, and makes the following contributions:

- We develop a modeling framework that allows us to represent the LRP as an integer multicommodity network flow problem with side constraints in a suitably defined graph. This graph corresponds to a two-layer time-space network that keeps track of the maintenance status of specific locomotives over the planning horizon, as well as managing the assignment of locomotives to trains based on their current maintenance status. We allow locomotives to miss their maintenance deadlines, for example due to insufficient shop capacity, while forbidding them to pull trains or to light travel until they have been serviced in a shop. This differs from the most restrictive assumption in the literature, where locomotives must be serviced punctually after a fixed number of operating days or after traveling a given number of miles.

- We propose a tractable integer linear programming (ILP) formulation for the LRP, which can be solved optimally by current state-of-the-art mixed integer programming (MIP) solvers within reasonable computing time. The size of the formulation depends on its underlying graph, which we keep within a manageable size by using commodity aggregation and flow decomposition techniques, and by considering only suitable subsets of repositioning and maintenance opportunities. For a typical one-week instance, our model has over 2.3 million constraints and 3.8 million integer variables, and can be optimally solved in less than 10 minutes, on average. Unlike previous solutions methodologies, we do not need to devise specialized algorithmic strategies to find optimal solutions in short computing times.

- We perform extensive computational experiments to assess the performance of the model and evaluate how variations on key parameters, such as shop capacity, connecting times and repositioning costs, affect the structure of optimal solutions. Our findings indicate that locomotive repositioning is sensitive to variations in repositioning costs, as one would expect, and that the weekly fleet size is largely impacted by variations in both connecting times and repositioning costs. Interestingly, reductions in shop capacity have a minor impact on the system performance, which suggests that it is well protected against major shop disruptions. Our methodology can be used to run multiple scenario analyses and support decision-making.

- We compare solutions obtained by our model with those implemented in practice by the company, and show that our formulation provides solutions that require both fewer locomotives and less repositioning. Our methodology, coupled with the models and algorithms developed by Ortiz-Astorquiza et al. (2019), can help the company manage its locomotive fleet in a more cost-effective way, while respecting relevant operational and business constraints. Note that although we focus on CN's case study, the maintenance rules and regulations we consider are applicable to the whole North American market. Hence, our methodology is applicable to other railway companies by adjusting the necessary side constraints.

The rest of the paper is organized as follows. Section 2 provides the problem description, while Section 3 describes in detail our time-space network representation. The mathematical formulation and computational experiments are reported in Sections 4 and 5, respectively. Conclusions and future research directions are discussed in Section 6.

2. Problem Description

This paper is based on the LRP currently faced by the CN, a class I North American railroad. In this problem we aim to determine the route followed by each locomotive over a one-week planning horizon such that the total operational cost is minimized while satisfying the power requirements of a given train schedule, balancing the locomotive flows, and respecting train connections and locomotive maintenance.

2.1. Problem Data

We now describe the input data required for the LRP studied in this paper. We assume that all the data is deterministic and known in advance.

Table 1 summarizes the notation of sets and parameters used in this section.

Train Schedule. It contains the set of trains that operate during the planning horizon. For each train l it defines a unique ID, a tonnage t_l, a horsepower per tonnage (HPT) factor β_l, and route information that specifies origin, destination, power changing stations, and their corresponding times of departure and arrival. In combination, ton_l and β_l allow us to calculate how much horsepower (HP) is needed to pull the train. Power changing stations are intermediate stations along the train route where it can add or drop locomotives. It is worth mentioning that some trains might be split into multiple legs in a pre-processing stage, depending on whether they need to change their consist at predefined power changing stations. The schedule provides all relevant information for each of the train legs.

Assignment Plan. Let $\mathcal{K}$ be the set of locomotive types. This plan specifies the number ρ_{kl} of locomotives of type $k \in \mathcal{K}$ that must be assigned to each train l in the schedule. For those trains that are split into multiple legs, the plan gives the corresponding assignment for each of them. Furthermore, it also specifies pairs of trains that must be assigned the same consist (i.e., a list Q of train-to-train connections).

Locomotive Data. Let $\mathcal{V}$ be the set of locomotives. For each locomotive $v \in \mathcal{V}$ we know its different attributes, such as ID, type (k_v), horsepower

Sets			
$\mathcal{K}$	locomotive types	Q	train-to-train connections
$\mathcal{V}$	all locomotives	$\mathcal{S}$	all stations
$\mathcal{V}^C$	critical locomotives	$\mathcal{S}^{SH}$	shop stations
$\mathcal{V}_k^C$	critical locomotives of type k	$\mathcal{M}$	maintenance types
Parameters			
ton_l	tonnage of train l	δ_v	maintenance deadline of critical locomotive v
β_l	horsepower per tonnage factor of train l	r_{ij}	railroad distance between stations i and j
ρ_{kl}	number of locomotives of type k required by train l	m_{ks}	number of locomotives of type k needed at station s by the end of the planning horizon
k_v	type of locomotive v		
h_v	horsepower of locomotive v		
w_v	weight of locomotive v	C_s	capacity of shop s
λ_v	number of axles of locomotive v	d_m	duration of maintenance of type m
m_v	type of maintenance required by critical locomotive v		

Table 1: Notation of data sets and parameters for the LRP.

(h_v), weight (w_v), number of axles (λ_v), status and location. The locomotive status indicates whether it is initially in maintenance, in transit in a train, or idling in a yard. The locomotive location is the station (i.e., yard or shop) where it is located at the beginning of the horizon. We also know the subset $\mathcal{V}^C \subseteq \mathcal{V}$ of locomotives with scheduled maintenance during the current horizon, referred to as *critical locomotives*. For each locomotive $v \in \mathcal{V}^C$, we know the specific type of maintenance that it requires, m_v, and its associated maintenance deadline, δ_v. Furthermore, let $\mathcal{V}_k^C \subseteq \mathcal{V}^C$ be the set of critical locomotives of type $k \in \mathcal{K}$.

Network Data. It specifies information on the railroad network, such as the set of railroad stations $\mathcal{S}$, railroad distance r_{ij} between stations $i, j \in \mathcal{S}$, and shop stations. For each station $s \in \mathcal{S}$, there is a unique ID and a minimum number m_{ks} of locomotives of type $k \in \mathcal{K}$ that must be made available at s by the end of the horizon. It corresponds to an estimate of the number of locomotives needed to meet the power demand at the beginning of the next planning horizon. The set of shop stations is denoted by $\mathcal{S}^{SH} \subseteq \mathcal{S}$. The capacity of shop $s \in \mathcal{S}^{SH}$, denoted by C_s, specifies the maximum number of locomotives in service at any time. The set of maintenance types is denoted by $\mathcal{M}$, and the duration of a maintenance of type $m \in \mathcal{M}$ is d_m.

Cost Data. It specifies different cost parameters such as track maintenance, fuel consumption, crew, maintenance and ownership costs. The track maintenance cost is associated to the usage of the railroad. The fuel consumption cost depends on the locomotive type, fuel consumption rate and distance. The crew cost is associated to the cost of operating a locomotive, and depends both on the time and distance traveled by the crew. The maintenance cost relates to the time required to service a locomotive, and depends on the locomotive and maintenance types. Finally, the ownership cost corresponds to the weekly value of owning a locomotive, and depends on factors such as locomotive type, acquisition value, lifetime, overhauls and residual value.

2.2. Problem Constraints

In this section, we present the operational and business constraints we must consider in order to solve the LRP.

Locomotive Flow Balance. An important aspect of locomotive route planning is to ensure that there are enough locomotives of the required types at the right stations to meet the train schedule. Thus, we must determine how to reposition locomotives to meet power requirements at the different stations. To accomplish this, we can make use of *deadheading* and *light traveling*. Deadheading locomotives do not pull a train. Instead, they are pulled like railcars by a set of active locomotives from one place to another. Light traveling locomotives reposition themselves between different stations, without pulling railcars. A set of locomotives in light travel forms a group, and one locomotive in the group pulls the other ones from an origin station to a destination station (Ahuja et al., 2005b; Vaidyanathan et al., 2008a). Notice that light traveling is not scheduled. Thus, it is more flexible than deadheading. However, it is also more expensive as it requires an additional crew to operate the pulling locomotive, and consumes track capacity that could otherwise be used by trains moving freight.

Locomotive Maintenance. Honoring locomotive maintenance represents a major challenge in locomotive route planning. Each unit is forced to pass through maintenance periodically (e.g., every 92 days in North America) for routine maintenance. Additionally, from time to time, they need to pass by a shop for other major revisions and mechanical repairs. If a locomotive misses a shop appointment it has to be turned off and deadheaded to a shop for maintenance (Bouzaiene-Ayari et al., 2016).

Train-to-train Connections. Whenever a train arrives at its destination, its consist can be either assigned in its entirety to a train departing later from the same station (referred to as a train-to-train connection), or dismantled (referred to as busted) so that each individual locomotive goes to a pool of locomotives from which new consists are formed (Ahuja et al., 2005a). Consist busting is undesirable as it requires additional locomotive and crew time to decouple and move locomotives individually. It might also result in delays as departing trains get their locomotives from several arriving trains, one of which might be delayed. Thus, at the tactical level, the LAP determines which train-to-train connections to make so as to reduce consist busting. When solving the LRP we must satisfy this predefined list of *train-to-train* connections.

Power Requirements and Train Capacity. We must assign locomotives of the required types to each train to satisfy power requirements.

The number of locomotives of each type assigned to a given train depends on the horsepower required to pull it, and is an output of the LAP. The consist assigned to a train can also include foreign locomotives, which may be necessary to cover locomotive unavailability due to maintenance. In addition, we must also respect limits on the number of locomotives attached to a train (both active and deadheading) or light traveling in a group.

3. Time-Space Network

We formulate the LRP based on a two-layer time-space network, which represents the physical railroad activities and events of interest over the planning horizon. Using multi-layer time-space networks as a modeling framework is a common practice in the rail transportation literature (see, for example, Zhu et al., 2014). Table 2 summarizes the main notation used throughout this section to describe our time-space network.

Nodes			
N_D^B	departure nodes in the service layer	N_F^T	final nodes in the overdue layer
N_A^B	arrival nodes in the service layer	N_{SH}^B	nodes that provide a maintenance opportunity in the service layer
N_O^B	outpost nodes in the service layer		
N_R^B	source nodes in the service layer	N_D	all departure nodes
N_I^B	initial nodes in the service layer	N_A	all arrival nodes
N_H^B	sink nodes in the service layer	N_O	all outpost nodes
N_F^B	final nodes in the service layer	N_R	all source nodes
N_Q^B	connection nodes in the service layer	N_I	all initial nodes
N_D^T	departure nodes in the overdue layer	N_H	all sink nodes
N_A^T	arrival nodes in the overdue layer	N_F	all final nodes
N_O^T	outpost nodes in the overdue layer	$\mathcal{N}^B$	all nodes in the service layer
N_R^T	source nodes in the overdue layer	$\mathcal{N}^T$	all nodes in the overdue layer
N_I^T	initial nodes in the overdue layer	$\mathcal{N}$	all nodes
N_H^T	sink nodes in the overdue layer	t_i, p_i	time and location of node i
Arcs			
A_T^B	train arcs in the service layer	$A_{SH}(v)$	all shop arcs for locomotive v
A_Q^B	train-to-train arcs in the service layer	A_{T-}^T	legacy train arcs in the overdue layer
A_L^B	light traveling arcs in the service layer	A_{SH-}^T	legacy shop arcs in the overdue layer
A_{DH}^B	deadheading arcs in the service layer	A_{B2T}	*B2T* inter-layer arcs
A_G^B	ground arcs in the service layer	$A_{B2T}(v)$	*B2T* inter-layer arcs for locomotive v
A_{SH}^B	shop arcs in the service layer	A_{T2B}	*T2B* inter-layer arcs
$A_{SH}^B(v)$	shop arcs for locomotive v in the service layer	$A_{T2B}(v)$	*T2B* inter-layer arcs for locomotive v
		$\mathcal{A}^B$	all arcs in the service layer
A_{T-}^B	legacy train arcs in the service layer	$\mathcal{A}^T$	all arcs the overdue layer
A_{SH-}^B	legacy shop arcs in the service layer	A_G	all ground arcs
A_{DH}^T	deadheading arcs in the overdue layer	A_{DH}	all deadheading arcs
A_G^T	ground arcs in the overdue layer	A_{SH}^-	all legacy shop arcs
A_{SH}^T	shop arcs in the overdue layer	A_T^-	all legacy train arcs
$A_{SH}^T(v)$	shop arcs for locomotive v in the overdue layer	$\mathcal{A}$	all arcs

Table 2: Notation of nodes and arcs in the time-space network.

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a graph where $\mathcal{N}$ denotes the set of nodes and $\mathcal{A}$ represents the set of arcs. Each node $i \in \mathcal{N}$ represents an event and is associated with two attributes: time (t_i) and location (p_i). Each arc $l \in \mathcal{A}$ represents an activity, such as pulling a train, deadheading, light traveling, waiting at a station, going to a shop for maintenance, or a train-to-train connection.

Although undesirable, in practice a locomotive might miss its shop appointment and be serviced after its deadline due to insufficient shop capacity. Such locomotives are said to be in *overdue* state. Our two-layer time-space network allows us to manage and separate the flows of overdue and non-overdue locomotives. The *service layer* $\mathcal{G}^B = (\mathcal{N}^B, \mathcal{A}^B)$, where $\mathcal{N}^B$ and $\mathcal{A}^B$ denote the set of nodes and arcs, respectively, contains arcs that represent all different locomotive activities, including pulling a train and light traveling. Only regular locomotives, those with no shop appointment, and critical locomotives that have not missed their deadline can flow on this layer. Overdue locomotives flow in the *overdue layer* $\mathcal{G}^T = (\mathcal{N}^T, \mathcal{A}^T)$, where $\mathcal{N}^T$ and $\mathcal{A}^T$ denote the set of nodes and arcs, respectively. This layer does not contain any arcs that represent pulling a train, light traveling, or making train-to-train connections. Whenever a critical locomotive misses its deadline, it is immediately transferred to the overdue layer, where it flows until passing through a shop. Afterward, the locomotive returns to the service layer, where it can again pull trains, light travel or make train-to-train connections.

It is worth noting that we could attain the same result without resorting to the utilization of a multi-layer structure. This would require, however, the inclusion of additional linking constraints to manage the assignment of a given critical locomotive to a train or light travel based on its current maintenance status (i.e., overdue or not). By using another layer, we avoid adding such linking constraints while meeting the operational requirement of not assigning an overdue locomotive to a train or light travel until it has been serviced in a shop. It also allows us to propose a mathematical formulation (see Section 4) whose structure can be exploited by state-of-the-art commercial solvers to provide optimal solutions within short computing times (see Section 5).

Next, we describe the different elements of our two-layer time-space network.

3.1. Service Layer

Figure 1 depicts an illustrative example of the service layer in a time-space network with three stations. We partition the set of nodes $\mathcal{N}^B$ into departure, arrival, outpost, connection, source, and sink nodes, respectively.

Departure (N_D^B) and Arrival (N_A^B) Nodes. Each $i \in N_D^B$ represents a train departure from its origin (white nodes in Figure 1). Its location attribute corresponds to the train origin station. Its time attribute is given by the train departure time minus the time required to build the consist. An arrival node $i \in N_A^B$ represents a train arrival at its destination (black nodes in Figure 1). Its location attribute corresponds to the train destination station, and its time attribute is given by the train arrival time plus the time required to bust the consist.

Outpost Nodes (N_O^B). We place these nodes at each station at different points in time, for example, at the beginning of each day or working shift. We use them to provide maintenance or light traveling opportunities at different stations (see dark gray nodes in Figure 1). We can think of them as events at specific points in time where it is necessary to make a decision, such as whether a locomotive should go to a shop, or light travel between two stations to reposition itself, or stay idle at its current location.

Connection Nodes (N_Q^B). Let $(q_1, q_2) \in Q$ represent a connection between trains q_1 and q_2, respectively. For each $(q_1, q_2) \in Q$ we create two nodes, say i and j. The location and time attributes of node i correspond to the destination and arrival time of train q_1. Conversely, the location and time attributes of node j correspond to the origin and departure time of train q_2. These two nodes are the end points of a specific type of arc that represents the consist transfer between trains q_1 and q_2, respectively. See light gray nodes in Figure 1. For the sake of clarity, we only depict a pair of connection nodes (i.e., only one train-to-train connection). The tail node, i, is to the left of the arrival node of train q_1, as the consist is not busted upon arrival. Likewise, the head node, j, is to the right of the departure node of train q_2, as the consist does not need to be built before departure.

Source Nodes (N_R^B). At each station $s \in \mathcal{S}$, we represent the beginning of the planning horizon with a special node $i \in N_R^B$ with location and time attributes set to s and 0, respectively (see hatched nodes at time 0 in Figure 1). We call this node the *initial node* of station s, and denote the set of all initial nodes in the service layer by N_I^B. Each $i \in N_I^B$ is a source

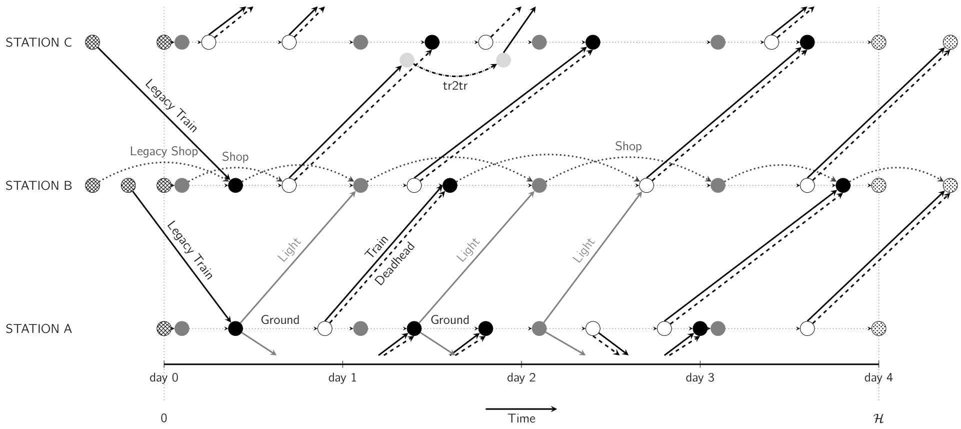


Figure 1: Example of the service layer in a time-space network with three stations.

of available locomotives at station $s = p_i$ at the beginning of the planning horizon. Each node $i \in N_R^B \backslash N_I^B$ represents an event that took place during the previous horizon (see hatched nodes to the left of time 0 in Figure 1). This event is the beginning of an activity that finishes within the current planning period. These nodes are sources of locomotives that are unavailable at the beginning of the horizon, such as those in transit or in maintenance at time 0.

Sink Nodes (N_H^B). Let $\mathcal{H}$ denote the end of the current planning horizon. At each station $s \in \mathcal{S}$, we represent the end of the planning horizon with a special node $i \in N_H^B$, with location and time attributes equal to s and $\mathcal{H}$, respectively (see dotted nodes at time $\mathcal{H}$ in Figure 1). We call this node the *final node* of station s, and denote the set of all final nodes in the service layer by N_F^B. Each $i \in N_F^B$ is a sink for locomotives available at station $s = p_i$ by the end of the planning horizon. Each node $i \in N_H^B \backslash N_F^B$ represents an event that will take place during the upcoming planning horizon, as depicted by dotted nodes to the right of time $\mathcal{H}$ in Figure 1. These nodes are sinks for locomotives that are in transit or in maintenance at time $\mathcal{H}$.

From now on, we assume that nodes at each station are sorted in chronological order by their time attribute, and that no pair of nodes at the same station has the same time attribute. We also partition the set of arcs $\mathcal{A}^B$ into different sets, namely, train, train-to-train, deadheading, light traveling, shop, ground and legacy arcs. We next describe each of these sets of arcs.

Train (A_T^B) and Train-to-Train (A_Q^B) Arcs. The set A_T^B consists of one arc l for every train in the schedule (solid black arcs in Figure 1). These arcs connect a departure node with its corresponding arrival node. If the time attribute of an arrival node is greater then $\mathcal{H}$, then we change it to a sink node. The set A_Q^B contains one arc for each train-to-train connection $(q_1, q_2) \in Q$ (dashdotted arc, named *tr*2*tr*, in the middle top part of Figure 1). Each $l \in A_Q^B$ links two connection nodes, i and j, associated to trains q_1 and q_2, respectively. Let l_1 and l_2 be the train arcs that represent trains q_1 and q_2, respectively. To enforce a train-to-train connection, we set node i as head of arc l_1, and node j as tail of arc l_2, respectively. Notice that the former head of l_1 is an arrival node, and the former tail of l_2 is a departure node. By changing these nodes we enforce the connection between the two trains: l_1 is the only incoming arc into node i, while l_2 is the only outgoing arc from node j. Since i and j are uniquely connected by a train-to-train arc, the consist assigned to l_1 is transferred to l_2.

Deadheading Arcs (A_{DH}^B)**.** For each train in the schedule there is an arc $l \in A_{DH}^B$ that represents a deadheading opportunity from the train origin to its destination (see dashed arcs in Figure 1). We also include arcs that represent deadheading opportunities (i) from the train origin to power changing stations in its route, (ii) from power changing stations in the train route to the train destination and, (iii) between power changing stations in the train route. For the sake of simplicity, Figure 1 does not show all deadheading options for each train. Arcs representing deadheading from the train origin must outbound from the corresponding train departure node. Conversely, arcs representing deadheading to the train destination must inbound at the corresponding train arrival node. Also, end points of arcs representing deadheading between power changing stations must respect arrival and departure times at the stations.

Light Traveling Arcs (A_L^B)**.** They are depicted with solid gray arcs in Figure 1. Including all possible light traveling options is impractical as it would make the problem computationally intractable for instances of realistic size. Thus, we only consider a suitable subset of light traveling arcs, which we generate following the procedure described in Section 3.4.

Shop Arcs (A_{SH}^B)**.** We consider different types of maintenance, each one with different frequency, duration and cost. Similar to light traveling arcs, including all possible options is impractical. Therefore, we consider only a reduced number of maintenance opportunities, as described in Section 3.5. For now, let $m(l)$ denote the type of maintenance associated to arc $l \in$

A_{SH}^B. Then, for each locomotive $v \in \mathcal{V}^C$, we define $A_{SH}^B(v) = \{l = (i,j) \in A_{SH}^B \mid m(l) = m_v, t_i \leq \delta_v\}$ as the set of shop arcs that can be taken by locomotive v in the service layer. In Figure 1, maintenance is represented by dotted gray arcs at the station B. For simplicity, we only depict one shop arc at each maintenance opportunity. In practice, however, we include one shop arc for each maintenance type.

Ground Arcs (A_G^B). Depicted by horizontal dotted black arcs in Figure 1, these arcs represent locomotives idling at a given station, waiting for upcoming trains, maintenance or light traveling opportunities. We recall that nodes at each station are assumed to be sorted chronologically by their time attribute. Thus, starting with the initial node we add a ground arc to connect each node with the next one in the sequence, until reaching the final node of the station. This allows us to model the flow of locomotives at a given station over time, from the beginning to the end of the planning horizon.

Legacy Arcs (A_{LEG}^B). This set represents activities that started during the previous planning horizon, but will finish within the current one. The set of *legacy train arcs* (A_{T-}^B) contains one arc for each train in the schedule of the previous horizon that reaches its destination within the boundaries of the current one (solid black arcs crossing time 0 in Figure 1). Its tail is a source node that represents the train departure during the previous horizon. Its head represents the train arrival at destination and is, therefore, an arrival node. The set of *legacy shop arcs* (A_{SH-}^B) contains one arc for each locomotive in maintenance at time 0 (dotted gray arcs crossing time 0 in Figure 1). Its tail is a source node that represents the beginning of the maintenance in the previous horizon. Its head denotes the end of the maintenance, and is the first node at the shop station with the proper time attribute. Flows on legacy train and legacy shop arcs are known in advance, as they correspond to decisions made in the previous horizon.

3.2. Overdue Layer

To create the overdue layer of our time-space network we initially make a copy of the service layer, described above. This way, each node in the service layer has exactly one copy in the overdue layer, with the same location and time attributes. We then remove from it all train, light traveling, and train-to-train arcs, which cannot be traversed by overdue locomotives. Let

N_D^T, N_A^T, N_R^T, N_I^T, N_H^T, N_F^T and N_O^T denote the sets of departure, arrival, source, initial, sink, final and outpost nodes in the overdue layer, respectively. Likewise, let A_{DH}^T, A_G^T, A_{SH}^T, A_{T-}^T and A_{SH-}^T denote the sets of deadheading, ground, shop, legacy train and legacy shop arcs in the overdue layer. We also define $A_{SH}^T(v) = \{l = (i,j) \in A_{SH}^T \mid m(l) = m_v, t_i > \delta_v\}$ as the set of shop arcs that can be taken by locomotive $v \in \mathcal{V}^C$ in the overdue layer.

Considering both layers, we define $N_D = N_D^B \cup N_D^T$, $N_A = N_A^B \cup N_A^T$, $N_O = N_O^B \cup N_O^T$, $N_R = N_R^B \cup N_R^T$, $N_I = N_I^B \cup N_I^T$, $N_H = N_H^B \cup N_H^T$, and $N_F = N_F^B \cup N_F^T$. Similarly, we define $A_G = A_G^B \cup A_G^T$, $A_{DH} = A_{DH}^B \cup A_{DH}^T$, $A_{SH}^- = A_{SH-}^B \cup A_{SH-}^T$, $A_{T-} = A_{T-}^B \cup A_{T-}^T$ and, for $v \in \mathcal{V}^C$, $A_{SH}(v) = A_{SH}^B(v) \cup A_{SH}^T(v)$.

3.3. Connection Between Service and Overdue Layers

We connect service and overdue layers, allowing the flow of critical locomotives from one to another. The set A_{B2T} contains all arcs that allow the flow from the service layer to the overdue layer. There is one arc for each station $s \in \mathcal{S}$ and each time period over the planning horizon. Each arc $l \in A_{B2T}$ connects a node representing the last event at a station in a given time period to its copy in the overdue layer. While in the service layer, each locomotive $v \in \mathcal{V}^C$ should be moved to a shop before the end of period $\hat{t}_v = \lceil \delta_v / \tau \rceil$, where τ denotes the duration of one period (e.g., a day). Otherwise, it must immediately be moved to the overdue layer by taking one arc $l \in A_{B2T}$ placed by the end of period $\hat{t}_v$. We define $A_{B2T}(v) = \{l = (i,j) \in A_{B2T} \mid \lceil t_i / \tau \rceil = \hat{t}_v\}$ as the set of *B2T* arcs that can be taken by locomotive $v \in \mathcal{V}^C$.

Conversely, the set A_{T2B} contains all arcs that allow the flow of locomotives from the overdue layer to the service layer. It has one arc for each shop arc in the overdue layer whose head is not a sink node. Each arc $l \in A_{T2B}$ connects the head j of an arc $l' = (i,j) \in A_{SH}^T$ to its copy in the service layer. Once in the overdue layer, locomotive $v \in \mathcal{V}^C$ can only return to the service layer after it has been serviced in a shop. Thus, we define $A_{T2B}(v) = \{l = (i,j) \in A_{T2B} \mid t_i > \delta_v + d_{m_v}\}$ as the set of *T2B* arcs that can be taken by locomotive v, where d_{m_v} denotes the duration of the maintenance required by locomotive v, m_v.

3.4. Generating Light Traveling Arcs

Next, we describe a procedure to generate a suitable subset of light traveling arcs. We based our choice of considering only a small subset of light traveling arcs on the fact that, in practice, railroads prefer not using light traveling to reposition locomotives as it is a costly practice. First, we build a space network, which corresponds to a complete graph where nodes represent train stations. Then, we solve a minimum cost flow problem to determine the optimal flow of power through this network. The supply or demand of each node is calculated as the difference between inbound and outbound horsepower. Sources are, therefore, stations that receive more horsepower than they need and, conversely, sinks are stations that receive less horsepower than they need. The objective function value coefficient for each arc (i, j) in the network is set as follows:

$$e_{ij} = \begin{cases} r_{ij} & \text{if } o_{ij} \leq 2 \\ r_{ij} \cdot \alpha & \text{if } 2 < o_{ij} < \alpha \\ r_{ij} \cdot \alpha^2 & \text{otherwise.} \end{cases} \tag{1}$$

The cost e_{ij} of each arc (i, j) depends on the number o_{ij} of trains operated between stations i and j, an input parameter α, used to discourage the flow between i and j if there is more than a given number of trains operated between them, and the railroad distance between stations, r_{ij}. The rationale behind it is that, in practice, repositioning locomotives from i to j can be done through deadheading, instead of light traveling.

We solve this minimum cost flow problem, and create light traveling arcs between a pair of stations if the optimal flow between them is above a given threshold value, θ. We add γ arcs emanating from arrival nodes at the origin station, and entering the first available node at the destination station after the corresponding travel time. If the number of arrival nodes at the origin station is greater than γ, we split the planning horizon into γ time windows of equal length and add one light traveling arc for each of them. If a certain time window does not include arrival nodes, we use the first available arrival node located at a neighbor time window. To guarantee that there exists at least one light traveling opportunity per day at the origin station, we also add arcs emanating from outpost nodes. Finally, we add arcs to connect stations with critical locomotives (available or arriving in legacy trains) to shop stations. These arcs, which also emanate from outpost nodes, provide opportunities to reposition critical locomotives directly from stations without maintenance capabilities to shops. In order to keep a reduced set of light traveling arcs, we can consider only a subset of shops. The choice of shops to consider may be based on distance, capacity or any other practical criterion.

3.5. Generating Shop Arcs

Next, we propose a procedure to generate a suitable subset of maintenance opportunities over the planning horizon, while guaranteeing that there exists at least one opportunity per shop per day. Recall that $\mathcal{M}$ denote the set of maintenance types. Each $m \in \mathcal{M}$ has different duration and cost. We provide one maintenance opportunity for each outpost node $i \in N_O^B$ located at a shop station. For each opportunity, we create one shop arc for each maintenance type $m \in \mathcal{M}$: the arc emanates from node i and enters at the first node j at the same station that satisfies $t_j \geq t_i + d_m$, where d_m denotes the duration of maintenance type m. Observe that the arc tail is always an outpost node, while its head may be a node of any type. Additionally, for each shop, we consider an extra maintenance opportunity associated to the last event (i.e., the last node of whatever type) of each period. This extra opportunity allows us to represent the move to the shop of any locomotive that has arrived at the station since the last opportunity associated to an outpost node. Following the same rationale, we create one shop arc for each maintenance type $m \in \mathcal{M}$. In this case, both the tail and head of the arcs may be nodes of any type. We add the tail of each shop arc to the set N_{SH}^B, which represents the set of nodes in the service layer that provide maintenance opportunities over the planning horizon.

4. Mathematical Formulation

In this section we provide a mathematical formulation built upon the two-layer time-space network representation described in Section 3. The problem is formulated as an integer linear programming (ILP) model, which corresponds to an integer multicommodity flow problem with side constraints, where locomotives represent the commodities flowing on the arcs of the graph. A formulation where each individual locomotive is considered a commodity flowing in the graph is presented in Appendix A. For instance sizes typically found in the railroad industry, with thousands of locomotives and trains operated per week, this formulation results in a large-scale optimization problem, which is difficult to solve optimally within reasonable computing time.

To circumvent this issue, and based on the observation that only critical locomotives need to be modeled individually so as to keep track of their maintenance status, we aggregate regular locomotives into types, as in the LAP. Thus, instead of having one commodity per locomotive, we consider one commodity per critical locomotive plus one commodity per locomotive type. This reduction from a few thousand commodities to only a few dozens has a dramatic impact on the computational resources needed to optimally solve the problem. More importantly, given that regular locomotives of the same type are essentially identical from a cost perspective (i.e., the cost of traversing an arc depends on the locomotive type, not in the locomotive itself), we also eliminate symmetric solutions without missing the true optimal solution of the problem.

An issue with this strategy is, however, that we need a post-processing step to decompose the aggregated flows of regular locomotives into individual locomotive paths. We can efficiently accomplish this task by means of a polynomial time flow decomposition algorithm (see Ahuja et al., 1993, for details). For each locomotive type $k \in \mathcal{K}$, the algorithm separates the arc flows into paths (our graph has no cycles), each one connecting a source node to a sink node. The regular locomotive v that traverses the identified path is one among those of type k that are located at the source of the path.

Consider the following additional notation:

Sets:

$I_k[i]$: Set of inbound arcs to node i that can be taken by regular locomotives of type k;

$O_k[i]$: Set of outbound arcs from node i that can be taken by regular locomotives of type k;

$I_v[i]$: Set of inbound arcs to node i that can be taken by critical locomotive v;

$O_v[i]$: Set of outbound arcs from node i that can be taken by critical locomotive v;

E^B_{il}: Set of arcs in the service layer that represent an "ongoing" deadheading in train l when it departs from the ith station in its route;

E^T_{il}: Set of arcs in the overdue layer that represent an "ongoing" deadheading in train l when it departs from the ith station in its route.

Parameters:

c_{kl}: Flow cost of a locomotive of type k on arc l;

c_{vl}: Flow cost of critical locomotive v on arc l;

ϕ: Unit penalty cost for not servicing a critical locomotive;

λ^R_{ki}: Supply of regular locomotives of type k at source i;

λ^C_{vi}: Supply of the critical locomotive v at source i;

η^R_{kl}: Flow of regular locomotives of type k on legacy arc $l \in A_{T^-} \cup A_{SH^-}$;

η^C_{vl}: Flow of critical locomotive v on legacy arc $l \in A^-_T$;

m^T: Maximum number of locomotives per train or light traveling group;

m^{DH}_l: Maximum number of deadheading locomotives in train l, $m^{DH}_l = m^T - \sum_{k\in\mathcal{K}} \rho_{kl}$;

n_l: Number of intermediate stops of the train l.

Decision Variables:

x_{kl} : Number of regular locomotives of type k that flow on arc l;

y_{vl} : 1 if critical locomotive v flows on arc l, 0 otherwise;

u_{ki} : Number of additional locomotives of type k supplied by source i.

The LRP is formulated as follows:

$$\begin{aligned} \min \sum_{k\in\mathcal{K}} \sum_{l\in A^B_T \cup A_{DH} \cup A^B_L \cup A_G \cup A^B_Q} c_{kl} \left(x_{kl} + \sum_{v\in\mathcal{V}^C_k} y_{vl} \right) + \sum_{v\in\mathcal{V}^C} \sum_{l\in A_{SH}(v)} c_{vl} y_{vl} \\ + \sum_{v\in\mathcal{V}^C} \phi \left(1 - \sum_{l\in A_{SH}(v)} y_{vl} \right) \end{aligned} \tag{2}$$

subject to:

$$x_{kl} = \eta^R_{kl}, \qquad k \in \mathcal{K}, l \in A^B_{T^-} \cup A^B_{SH^-} \tag{3}$$

$$y_{vl} = \eta^C_{vl}, \qquad v \in \mathcal{V}^C, l \in A^-_T \tag{4}$$

$$\sum_{l\in O_k[i]} x_{kl} = \lambda^R_{ki} + u_{ki}, \qquad k \in \mathcal{K}, i \in N^B_I \tag{5}$$

$$\sum_{l\in O_v[i]} y_{vl} = \lambda_{vi}^C, \qquad v\in\mathcal{V}^C, i\in N_I \tag{6}$$

$$\sum_{i\in N_H^B}\sum_{l\in I_k[i]} x_{kl} = \sum_{i\in N_R^B}\sum_{l\in O_k[i]} x_{kl}, \qquad k\in\mathcal{K} \tag{7}$$

$$\sum_{i\in N_H}\sum_{l\in I_v[i]} y_{vl} = 1, \qquad v\in\mathcal{V}^C \tag{8}$$

$$\sum_{l\in I_k[i]} x_{kl} = \sum_{l\in O_k[i]} x_{kl}, \qquad k\in\mathcal{K}, i\in\{N_D^B\cup N_A^B\cup N_O^B\cup N_Q^B\} \tag{9}$$

$$\sum_{l\in I_v[i]} y_{vl} = \sum_{l\in O_v[i]} y_{vl}, \qquad v\in\mathcal{V}^C, i\in\{N_D\cup N_A\cup N_O\cup N_Q^B\} \tag{10}$$

$$x_{kl} + \sum_{v\in\mathcal{V}_k^C} y_{vl} = \rho_{kl}, \qquad k\in\mathcal{K}, l\in A_T^B \tag{11}$$

$$\sum_{l\in A_{SH}(v)} y_{vl} \le 1, \qquad v\in\mathcal{V}^C \tag{12}$$

$$\sum_{l\in A_{B2T}(v)} y_{vl} = 1 - \sum_{l\in A_{SH}^B(v)} y_{vl}, \qquad v\in\mathcal{V}^C \tag{13}$$

$$y_{vl'} = \sum_{l=(i,j)\in A_{SH}^T(v)} y_{vl}, \qquad v\in\mathcal{V}^C, l'=(j,h)\in A_{T2B}(v) \tag{14}$$

$$\sum_{v\in\mathcal{V}^C}\sum_{\substack{l=(j,h)\in A_{SH}(v):\\ p_j=s\\ t_j\le t_i<t_h}} y_{vl} \le C_s - \sum_{\substack{l=(j,h)\in A_{SH}^-:\\ p_h=s\\ t_j\le t_i<t_h}}\sum_{k\in\mathcal{K}} \eta_{kl}^R, \qquad \begin{array}{l} s\in\mathcal{S}^{SH},\\ i\in N_{SH}^B:\\ p_i=s \end{array} \tag{15}$$

$$\sum_{k\in\mathcal{K}}\sum_{j\in E_{il}^B}\left(x_{kj} + \sum_{v\in\mathcal{V}_k^C} y_{vj}\right) + \sum_{v\in\mathcal{V}^C}\sum_{j\in E_{il}^T} y_{vj} \le m_l^{DH}, \qquad \begin{array}{l} l\in A_T^B,\\ i=0,\dots,n_l \end{array} \tag{16}$$

$$\sum_{k\in\mathcal{K}} x_{kl} + \sum_{v\in\mathcal{V}^C} y_{vl} \le m^T, \qquad l\in A_L^B \tag{17}$$

$$\sum_{\substack{i\in N_F^B:\\ p_i=s}}\left(\sum_{l\in I_k[i]} x_{kl} + \sum_{v\in\mathcal{V}_k^C}\sum_{l\in I_v[i]} y_{vl}\right) \ge m_{ks}, \qquad k\in\mathcal{K}, s\in\mathcal{S} \tag{18}$$

$$x_{kl}\in\mathbb{Z}^+, \qquad k\in\mathcal{K}, l\in\mathcal{A}^B\backslash A_{SH}^B \tag{19}$$

$$y_{vl}\in\{0,1\}, \qquad v\in\mathcal{V}^C, l\in\mathcal{A}^B\cup\mathcal{A}^T\cup A_{B2T}\cup A_{T2B} \tag{20}$$

$$u_{ki}\in\mathbb{Z}^+, \qquad k\in\mathcal{K}, i\in N_I^B. \tag{21}$$

The objective function (2) aims to minimize the total operational cost. The first term of (2) includes costs of pulling trains, deadheading, light traveling, idling at stations, and enforcing train-to-train connections. The second term is the maintenance cost and, finally, the last term is a penalty cost incurred for each critical locomotive not serviced in a shop by the end of the planning horizon. The cost of pulling a train is a function of track maintenance, fuel consumption and ownership costs. The deadheading cost is a function of track maintenance and ownership costs. Light traveling costs include track maintenance, ownership, fuel consumption and crew costs. We follow Ortiz-Astorquiza et al. (2019) and include fixed crew and fuel consumption costs within c_{kl} to penalize and discourage the use of light traveling arcs. Idling costs correspond exclusively to ownership costs. Similarly, train-to-train connection costs are associated to having the locomotives inactive while the connection takes place. The maintenance cost is a function of maintenance type and ownership costs. The cost of operating foreign locomotives is implicitly accounted for via ownership costs, which are paid on every arc of the graph. Thus, the railroad incurs all the costs of owning foreign units during the whole planning horizon. We note that we do not consider fixed leasing cost as long-term agreements between major railroads govern the availability and access to locomotives when required.

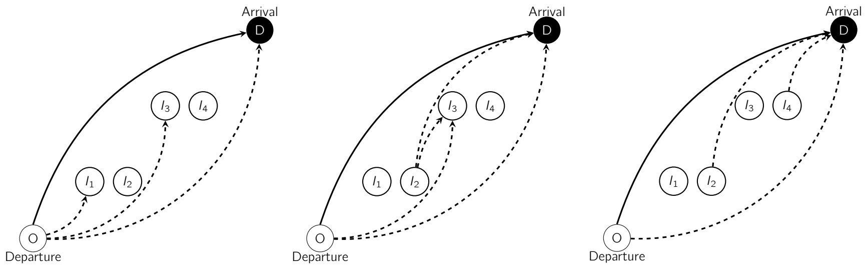
Constraints (3) and (4) set the initial conditions at the beginning of the week: the set of constraints (3) fixes the known flow of regular locomotives on legacy arcs. Observe that it suffices to consider only arcs in the service layer, as regular locomotives do not flow on the overdue layer. Similarly, constraints (4) set the known flow of critical locomotives on legacy train arcs. These are units initially in transit that are due for maintenance during the current week.

The set of constraints (5) establishes the number of regular locomotives of type k flowing out of the initial node i, considering both owned and foreign locomotives of type k. The integer variable u_{ki} states how many additional locomotives of type k are needed at station $s = p_i$ at the beginning of the planning horizon. Similarly, constraints (6) state the flow out of initial nodes for critical locomotives. In this case, λ^C_{vi} equals one if the critical locomotive v is available at station $s = p_i$ at the beginning of the horizon, or zero otherwise. Constraints (7) guarantee that the number of regular locomotives flowing into sink nodes equals the one flowing out of sources.

Likewise, constraints (8) indicate that each critical locomotive must flow into a sink. Flow conservation on departure, arrival, outpost and connection nodes are imposed by constraints (9), for regular locomotives, and by (10), for critical locomotives, respectively.

Constraints (11) guarantee that each train is assigned the proper type and number of active locomotives, considering critical, regular, and foreign locomotives, if needed. Constraints (12) guarantee that each locomotive $v \in \mathcal{V}^C$ flows, at most, on one shop arc in the set $A_{SH}(v)$, which contains only the shop arcs that can be traversed by locomotive v. Constraints (13) force critical locomotives to flow toward the overdue layer if they miss their maintenance deadline. Constraints (14) establish that overdue locomotives flow back to the service layer as soon as they have been serviced in a shop. Constraints (15) correspond to shop capacity constraints, which impose a limit on the number of locomotives in a given shop s during each possible maintenance opportunity. For each opportunity $i \in N^B_{SH}$ at shop s, the left-hand side of (15) calculates the flow on all shop arcs at s that represent an ongoing maintenance operation at the time t_i (i.e., the number of locomotives in shop s at the time t_i). The right-hand side corresponds to the capacity of shop s at time t_i, considering any locomotive initially in shop that is still under inspection at the time t_i.

Constraints (16) limit the number of deadheading locomotives attached to a train when it departs from the ith station in its route. Figure 2 provides an example of a train with two intermediate stops at power changing stations. The train is represented by the solid arc, while dashed arcs represent the deadheading opportunities along the route. Similarly, constraints (17) impose a limit on the number of locomotives traversing a light traveling arc.



(a) Departure from O. (b) Departure from I_2. (c) Departure from I_4.

Figure 2: Example of deadheading arcs for a train with two intermediate power changing stations.

Constraints (18) enforce a minimum number of locomotives of each type at each station by the end of the planning horizon. Observe that we only consider locomotives flowing into final nodes, as those units are available to be used at the beginning of the next week. Locomotives flowing into other sinks are either in transit or in maintenance, and cannot be used right away to provide power to any train.

5. Computational Experiments

In this section we perform computational experiments to asses the performance of our methodology on a set of real instances. We also conduct different scenario analyses and draw insights from them. We implemented all our algorithms in C++ and run them on a 2.40 GHz Intel Gold 6148 Skylake processor with 20 GB of memory. We modeled our ILP using the IBM Concert Technology and solved it using the CPLEX 12.10 solver with a single-thread and a one-hour computing time limit. In addition, we used the CPLEX Network Optimizer to solve the minimum cost flow problem described in Section 3.4 to generate light traveling arcs.

5.1. Benchmark Instances

We generated a set of 51 weekly instances based on CN's historical data, which includes the actual consist type assigned to each train. The network has over 1,700 stations out of which 480 act as train origins or destinations, and 19 can provide maintenance services. Shop capacity ranges from 1 locomotive for the smallest shops to 21 locomotives for the largest ones. A typical one-week train schedule has over 3,800 trains, which depart from or arrive at 373 different stations. The nominal fleet is composed of 2,205 locomotives, classified into five different types based on their operational characteristics. Every week, on average, 89% of the fleet (i.e., 1,958 locomotives) is actually available due to major unscheduled repairs and leasing to other railroads. Moreover, from the actually available fleet, on average, 91 locomotives are due for maintenance every week. Thus, our time-space graph has 96 commodities and results in a formulation with 2.3 million constraints and 3.9 million integer variables, on average. We use this set of real instances for our computational experimentation in Sections 5.3 and 5.4. In Section 5.2, we consider a set of smaller instances to showcase the impact of our aggregation strategy.

It is important to highlight that the data correspond to CN's actual operations, which reflect decisions made to handle unforeseen real-time events, such as unscheduled locomotive repairs. Therefore, there was a significant challenge involved in going from raw operational data to a clean version that could be used to generate instances for our optimization model. Along this research, CN's personnel assisted us to validate both data and solutions. Also, we set model parameters based on extensive preliminary experiments and CN's input. In particular, we empirically set light traveling related parameters (Section 3.4) to closely reproduce historical data. This required striking a balance between having a suitable set of high-quality light traveling opportunities and a manageable problem size. It also required to properly approximate light traveling costs, which generally include subjective decision-maker preferences. All cost parameters were estimated according to the actual values and guidelines followed by the company in practice.

Figure 3 reports the distribution of the number of constraints, total and binary variables of the model across the instances. Note that whiskers extend from the 5th to the 95th percentiles. As observed, our optimization problem is of large scale, with 95% of the instances having over 1.9 million constraints and 3.2 million variables. Most variables are binary and are associated to a small set of critical locomotives (91, on average). This provides a good idea of why considering individually each locomotive in the fleet is computationally intractable.

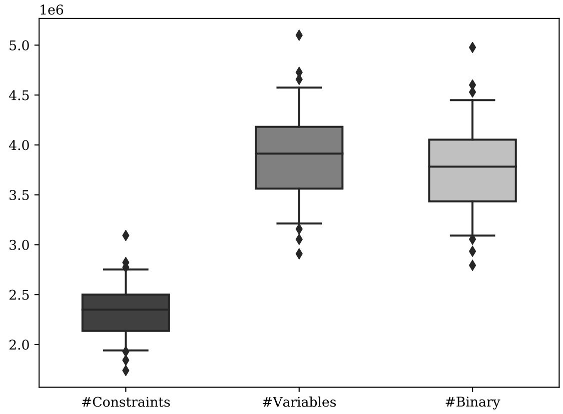


Figure 3: Distribution of the number of constraints, total and binary variables for our 51 instances.

5.2. The Value of Aggregation

In this section we assess the impact of our aggregation strategy on the computational performance of our methodology. In particular, we compare the formulation proposed in Section 4, where regular locomotives are aggregated into types, with the one in Appendix A, where regular locomotives are modeled individually. Initially, we tried solving real-size instances with both formulations. However, the dissagregated model could not solve such instances within a computing time limit of 24 hours. Therefore, in this section, we report experiments on smaller instances obtained by considering only mainline trains (i.e. heavy, long distance trains). These instances, while significantly smaller than the original ones, are still of realistic size (over 1500 trains per week, on average).

Table 3 summarizes the computational performance of both formulations in terms of number of constraints (#Cons.), total variables (#Vars.), binary variables (#Bins.), nodes explored in the branch-and-bound tree (#Nodes), optimality gap (Gap[%]), and computing time (CPU[min]). Observe that all results are aggregated by month.

Based on Table 3 we observe that aggregating regular locomotives into types has a significant impact on the solver's performance. Solving the aggregated formulation is more efficient due to the smaller size of the resulting problem and the streamlined space of feasible solutions that results from the elimination of symmetric solutions. Therefore, being able to solve real instances with the disaggregated formulation is unlikely due to the considerable amount of memory and time that it requires. As we show in the following sections, by using an aggregation strategy and applying flow decomposition in a post-processing step, optimal solutions for real instances

Month	Disaggregated Formulation						Aggregated Formulation					
	#Cons.†	#Vars.†	#Bins.†	#Nodes	Gap(%)	CPU(min)	#Cons.†	#Vars.†	#Bins.†	#Nodes	Gap(%)	CPU(min)
1	12.47	24.38	24.32	0	0.00	136.15	1.13	1.94	1.88	0	0.00	3.47
2	12.58	24.85	24.79	5	0.00	300.37	1.16	2.00	1.94	110	0.00	4.83
3	12.37	23.96	23.90	158	0.00	213.16	1.26	2.13	2.06	158	0.00	5.71
4	12.73	25.17	25.11	143	0.00	281.74	1.13	1.94	1.88	727	0.00	4.74
5	12.58	24.67	24.61	111	0.00	205.97	1.18	2.01	1.95	193	0.00	4.40
6	12.35	24.35	24.29	0	0.00	149.97	1.16	2.00	1.93	0	0.00	4.04
7	12.46	24.02	23.96	49	0.00	178.80	1.30	2.19	2.13	0	0.00	4.57
8	12.70	24.99	24.92	0	0.00	127.19	1.13	1.93	1.87	0	0.00	4.35
9	12.48	24.60	24.54	2	0.00	139.13	1.28	2.19	2.12	90	0.00	3.63
10	12.92	25.29	25.22	0	0.00	161.80	1.30	2.22	2.15	0	0.00	4.08
11	13.06	25.86	25.80	201	0.00	292.29	1.23	2.13	2.06	201	0.00	4.51
12	13.66	26.98	26.91	11	0.00	278.46	1.18	2.04	1.97	0	0.00	3.26
	12.68	24.90	24.84	59	0.00	206.03	1.20	2.06	1.99	126	0.00	4.34

†Values in millions.

Table 3: Comparison between disaggregated and aggregated formulations.

can be attained in very short computing times.

5.3. Comparison with CN's Operations

In this section we solve the set of real instances (Section 5.1) and compare CPLEX optimal solutions with those implemented by CN to show the potential savings that our optimization approach can achieve when planning locomotive routes. For confidentiality reasons, we do not show the actual cost of the solutions obtained by our methodology nor those implemented in practice. Thus, we resort to alternative statistics to measure solution quality and savings. Table 4 summarizes the relative difference of the optimal solutions with respect to CN actual operations, aggregated by month. Each row in the first four columns corresponds to the average over a month, and shows, in order, the relative difference in number of locomotives deadheading (Δ_{DH}), deadheading distance (Δ_{DH}^{D}), number of locomotives light traveling (Δ_{LT}), and light traveling distance (Δ_{LT}^{D}), respectively. Then, for each month, we show the minimum (maximum) relative difference for the number of owned, foreign, and total locomotives used, denoted by $\Delta_{O}^{\min}$ ($\Delta_{O}^{\max}$), $\Delta_{F}^{\min}$ ($\Delta_{F}^{\max}$), and $\Delta_{L}^{\min}$ ($\Delta_{L}^{\max}$), respectively. Finally, the last three columns show the average difference in total distance (Δ^{D}), the integrality gap between the initial linear programming relaxation and the optimal integer solution, $Gap^{0}(\%)$, and the computing time, CPU(min). The last row provides the average, minimum or maximum values over the 51 instances.

In terms of computing effort, despite their large size, all instances are optimally solved within short computing times (8.5 minutes, on average). In particular, the computing time ranges from 6 to 20 minutes, with 95% of the instances being solved in less than 15 minutes. Interestingly, about 90%

Month	Δ_{DH}	Δ_{DH}^{D}	Δ_{LT}	Δ_{LT}^{D}	$\Delta_{O}^{\min}$	$\Delta_{O}^{\max}$	$\Delta_{F}^{\min}$	$\Delta_{F}^{\max}$	$\Delta_{L}^{\min}$	$\Delta_{L}^{\max}$	Δ^{D}	$Gap^{0}(\%)$	CPU(min)
1	-11.82	36.77	-7.72	-47.00	-6.69	-5.06	-0.44	19.02	-4.68	-3.27	2.90	0.11	8.18
2	-23.23	6.75	20.26	-21.27	-7.29	-6.15	-2.63	17.81	-6.14	-3.80	2.44	0.22	9.30
3	-11.56	30.45	17.88	-19.28	-7.47	-2.55	-1.09	17.75	-4.99	-2.33	3.38	0.05	8.21
4	-14.31	10.66	-14.54	-55.81	-8.28	-4.46	-7.65	15.44	-8.17	-4.13	2.53	0.02	6.86
5	-7.61	26.09	-37.68	-63.64	-8.53	-6.26	-5.24	19.94	-6.65	-3.91	3.45	0.02	7.69
6	-20.63	-9.72	7.82	-38.85	-10.66	-7.25	-25.55	11.11	-10.65	-5.67	0.89	0.01	7.52
7	-18.74	9.73	-13.35	-50.49	-9.51	-6.39	-10.05	-2.72	-8.92	-5.91	2.62	0.10	8.38
8	-24.71	5.20	-18.31	-43.32	-10.11	-8.50	-14.42	-3.07	-10.42	-8.35	1.97	0.30	9.86
9	-28.35	-11.61	19.38	-27.11	-11.05	-7.96	-17.88	-6.80	-10.72	-10.09	0.81	0.03	8.60
10	-13.21	13.12	-13.38	-24.31	-9.51	-7.75	-13.67	1.65	-9.19	-7.19	2.93	0.12	12.80
11	-15.99	1.49	-29.34	-54.03	-10.32	-6.75	-11.08	1.94	-9.06	-4.97	1.48	0.02	8.59
12	-25.37	-3.76	9.93	-20.53	-9.84	-8.24	-13.48	1.89	-10.34	-6.19	0.82	0.01	6.27
	-17.81	9.88	-6.10	-39.69	-11.05	-2.55	-25.55	19.94	-10.72	-2.33	2.22	0.09	8.59

Table 4: Savings of the optimization approach in comparison to CN's actual operations.

of the instances are solved at the root node of the branch-and-bound tree. The average gap between the initial linear programming relaxation and the optimal integer solution is only 0.09%, which suggests that the formulation provides strong lower bounds and only few nodes need to be explored to find optimal solutions. Such strong bounds can be attributed to the structure of the problem, which is by design close to a multicommodity flow problem.

Besides its strong lower bounds, the good performance of our formulation can also be explained by the aggregation of individual regular locomotives into types, which eliminates a large number of symmetric solutions and reduces significantly the space of feasible solutions. Obviously, aggregation also allows us to have a smaller formulation in terms of both constraints and variables, which decrease considerably the computing time and memory required to solve the problem.

Overall, our formulation is able to obtain significant savings in terms of both locomotive repositioning and locomotive utilization, at the expense of traveling slightly longer distances. In terms of repositioning, our model provides optimal solutions with about 18% and 6% less deadheading and light traveling than CN, respectively. There is a 10% increase in the total deadheading distance, which is compensated by a 40% reduction of the total light traveling distance. We claim that reducing the light traveling distance to such a large extent compensates the increase in the deadheading distance, as in practice the former is more expensive than the latter. Indeed, light traveling implies using track capacity and crew time unproductively since no freight is moved and, consequently, no revenue is generated. Thus, keeping light traveling as low as possible means that both crew time and track capacity can be used in a more efficient way. Similarly, one can argue that moving deadheading locomotives for longer distances is less costly as the track and the crew have in any case to be used to operate the train to which the deadheading locomotive is attached to.

With respect to locomotive utilization, our formulation provides optimal solutions that require 2–11% fewer locomotives. This represents a significant extra buffer of locomotives that are available in the yards, which can be used to replace locomotives requiring unscheduled repair over the week. In terms of foreign locomotives, differences range from -25% to +20%. This is due to the current cost structure of the problem, which emphasizes the minimization of light traveling locomotives over the utilization of foreign units. Indeed, long-term agreements between major railroads make it easier and more cost effective to get extra power from other railroads instead of repositioning owned locomotives over long distances. Nevertheless, our model provides optimal solutions with less utilization of foreign locomotives for 30 out of our 51 instances.

5.4. Scenario Analysis

In this section we study three scenarios to illustrate how our methodology is valuable to analyze the impact of relevant operational events on the overall system performance. We first consider the impact of closing a major shop, so that shop capacity is considerably reduced. We then study the impact of longer connecting times on the locomotive utilization. Finally, we consider the trade-off between locomotive repositioning and fleet size. In all cases we use the optimal solutions reported in Section 5.3 as a baseline scenario.

5.4.1. Reduced Shop Capacity

Computational results for the baseline scenario indicate that the average maintenance capacity utilization is 62%, 80% and 80% for small, medium and large shops, respectively. Thus, a natural question to ask is whether the system is sensitive to drops in shop capacity, and whether our model can leverage routing and repositioning decisions in such situation. Therefore, in this section we consider the alternative scenario where the capacity of the largest shop in the system is set to zero. Table 5 shows the relative difference of the optimal solutions with respect to our baseline. All columns have the same meaning as in Table 4.

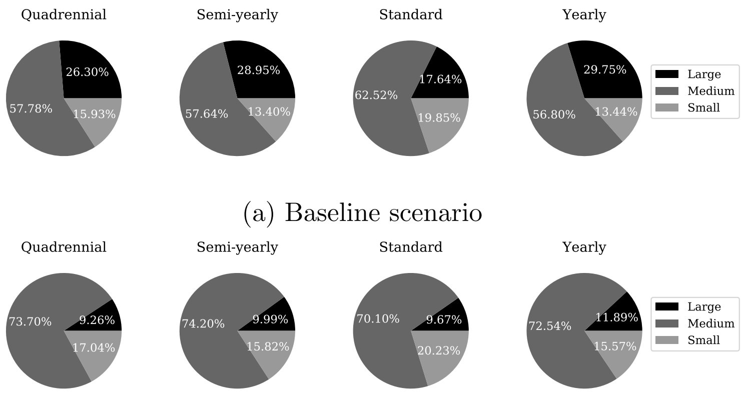
Overall, the effect on the number of owned, foreign and total locomotives required to satisfy the schedule is small, with deviations between -0.34–0.74%, -0.45–0.85% and -0.28–0.63%, respectively. This means that the company can still operate the schedule with a small increase in the number of required locomotives. The impact of locomotive repositioning is larger, but still reasonable. There is only a 2.30% increase in the number of deadheadings, while the number of light travels presents an increase of only 0.31%, on average. This increase in locomotive repositioning can be explained by the fact that some large shops are also major yards, with a high inbound traffic that is conveniently used to move critical locomotives to shop. Closing the largest shop then means that critical locomotives must utilize more deadheading and light traveling to find their way to alternative

Month	Δ_{DH}	Δ_{DH}^{D}	Δ_{LT}	Δ_{LT}^{D}	$\Delta_{O}^{\min}$	$\Delta_{O}^{\max}$	$\Delta_{F}^{\min}$	$\Delta_{F}^{\max}$	$\Delta_{L}^{\min}$	$\Delta_{L}^{\max}$	Δ^{D}
1	2.13	1.83	-1.35	-0.80	0.07	0.65	0.00	0.85	0.17	0.62	0.12
2	3.01	4.32	0.52	1.83	0.13	0.60	-0.45	0.39	0.11	0.51	0.25
3	2.70	2.76	-0.66	-1.97	-0.34	0.33	-0.37	0.00	-0.28	0.22	0.18
4	2.54	2.74	2.85	13.20	-0.20	0.66	-0.34	0.00	-0.17	0.50	0.17
5	1.01	3.28	2.60	1.30	-0.07	0.54	-0.37	0.00	-0.11	0.45	0.22
6	2.69	3.26	-1.92	-2.47	-0.14	0.74	0.00	0.31	-0.06	0.63	0.20
7	1.80	3.18	1.10	-0.10	0.14	0.60	-0.32	0.28	0.06	0.49	0.19
8	1.88	2.34	-1.03	0.62	0.07	0.48	-0.27	0.58	0.00	0.50	0.14
9	3.83	5.06	-1.36	-1.65	-0.07	0.62	0.00	0.54	0.00	0.54	0.29
10	2.57	5.04	-0.81	0.69	-0.07	0.62	-0.25	0.24	-0.11	0.48	0.29
11	1.56	3.02	3.27	1.59	-0.13	0.54	-0.24	0.25	-0.16	0.43	0.18
12	2.24	3.80	-0.76	0.71	0.20	0.54	-0.23	0.82	0.20	0.59	0.21
	2.30	3.37	0.31	1.11	-0.34	0.74	-0.45	0.85	-0.28	0.63	0.20

Table 5: Relative deviation with respect to the optimal solutions of the baseline for the scenario with reduced capacity.

shop locations. In addition, reducing shop capacity results in a 3.85% decrease in the number of critical locomotives serviced on time, as they have to wait longer or travel longer distances to find a spot in a shop. This, in turn, means more deadheading in the system as overdue locomotives cannot pull trains nor light travel between stations.

In terms of capacity, as one would expect, we observe an increase in shop utilization. The average capacity utilization goes from 62%, 80% and 80% for small, medium and large shops, to 64%, 87% and 84%, respectively. Figures 4a and 4b show the distribution of maintenance types across shops of different sizes. In the baseline scenario, maintenance is carried out mainly at medium and large shops, which account for more than 80% of the services in all cases. With the exception of the standard maintenance, large shops process at least 25% of the workload. This share drops to about 10% in the scenario where the largest shop is closed, which suggests that the remaining large shops are unable to absorb all the workload previously allocated to them. Indeed, most of that workload is redistributed to medium shops, which handle more than 70% of the maintenance services. On the one hand, this suggests that large shops are working close to their maximum capacity and cannot handle a significant amount of extra work. On the other hand, these results also suggest that the system is protected against major shop disruptions due to the spare capacity available at medium shops. Moreover, redistributing maintenance requests to other shops entails a very small increase in locomotive repositioning and utilization.



(a) Baseline scenario

(b) Reduced capacity scenario

Figure 4: Comparison of shop service distribution per maintenance type.

5.4.2. Longer Connecting Times

The time to build and bust consists, and more generally the time to maneuver locomotives upon arrival at yards, is an important parameter in our time-space graph that directly affects locomotive availability at stations. In this section we double connecting times at each station and analyze the effect of longer yard operations on the overall performance of the system. Table 6 shows the results of our experiments.

As observed, increasing connecting times has a negative effect in locomotive repositioning and, more significantly, in locomotive utilization. Longer

Month	Δ_{DH}	Δ_{DH}^{D}	Δ_{LT}	Δ_{LT}^{D}	$\Delta_{O}^{\min}$	$\Delta_{O}^{\max}$	$\Delta_{F}^{\min}$	$\Delta_{F}^{\max}$	$\Delta_{L}^{\min}$	$\Delta_{L}^{\max}$	Δ^{D}
1	2.25	0.33	7.78	3.93	-0.20	0.33	2.66	5.08	0.23	0.79	0.02
2	2.69	3.18	-6.55	-4.24	-0.20	0.80	0.73	8.14	0.73	1.47	0.19
3	1.11	-0.68	-4.73	-3.67	-0.20	0.00	3.66	5.88	0.44	0.85	-0.02
4	0.31	-0.09	1.77	6.17	-0.27	0.60	4.39	5.73	0.51	1.22	-0.01
5	-1.15	-2.02	5.70	2.66	-0.20	0.20	1.88	9.09	0.17	1.46	-0.14
6	2.46	2.06	-4.00	-5.47	-0.07	0.27	2.66	6.67	0.45	1.14	0.13
7	0.64	1.52	4.55	3.12	0.00	0.40	1.96	6.76	0.43	1.58	0.09
8	-0.39	-0.64	0.90	-2.33	-0.33	0.61	4.00	4.90	0.64	1.44	-0.05
9	1.66	2.86	1.31	-0.09	-0.13	0.28	3.36	4.84	0.76	1.13	0.17
10	0.53	0.76	3.53	0.59	-0.14	0.14	2.79	5.35	0.74	1.06	0.04
11	-0.33	1.08	6.69	2.99	0.07	0.66	2.92	6.67	0.91	1.52	0.09
12	0.19	-0.12	0.23	0.09	0.00	0.60	1.06	3.28	0.48	1.12	-0.02
	0.79	0.68	1.47	0.27	-0.33	0.80	0.73	9.09	0.17	1.58	0.04

Table 6: Relative deviation with respect to the optimal solutions of the baseline for the scenario with longer connecting times.

connecting times mean that locomotives must spend more time grounded, waiting longer for equipment and crew to maneuver them upon arrival, or simply waiting longer for consists to be assembled or dismantled. Since trains must be operated punctually, additional repositioning and extra locomotives are required to meet the schedule.

In particular, we observe an increase of up to 9% in the utilization of foreign locomotives, while deadheading and light traveling only increase by 0.79% and 1.47%, on average, respectively. Deadheading is cheaper than light traveling, but it is subject to limitations imposed by the train schedule, such as departure times, train routes and predefined power changing stations. Light traveling is more flexible, in the sense that it does not depend on the train schedule and one can decide where and when to use them, but is much more expensive. Given these operational constraints and the current cost structure of the problem, using extra foreign locomotives instead is a more convenient alternative.

5.4.3. Reduced Repositioning Costs

In practice, the light traveling cost is high, in comparison to the deadheading cost, to reflect the decision-maker preference of using as few light travels as possible. In this section, we gradually reduce the cost of light traveling and analyze how cheaper repositioning costs affect locomotive utilization across the network. This scenario represents a situation where the decision-maker is willing to accept a larger number of light traveling locomotives in the system in the expectation of achieving a better overall performance. Table 7 summarizes the results of the experiments, where each row corresponds to aggregated results for a given reduction percentage.

Reducing light traveling costs has a clear effect on the total number of locomotives required to meet train schedules. Intuitively, since repositioning locomotives is less expensive, light traveling becomes a convenient way of moving power (i) from stations with a surplus to others with a short-

Reduction(%)	Δ_{DH}	Δ_{DH}^{D}	Δ_{LT}	Δ_{LT}^{D}	$\Delta_{O}^{\min}$	$\Delta_{O}^{\max}$	$\Delta_{F}^{\min}$	$\Delta_{F}^{\max}$	$\Delta_{L}^{\min}$	$\Delta_{L}^{\max}$	Δ^{D}
10	5.06	2.79	59.52	15.27	-0.13	0.93	-14.44	-2.15	-1.86	-0.21	0.52
20	7.49	5.19	83.02	17.64	-0.20	1.52	-19.86	-2.41	-2.70	-0.48	0.84
30	9.07	6.24	107.33	22.14	0.00	2.02	-22.38	-3.74	-2.76	-0.59	1.03
40	9.24	6.36	122.32	22.26	0.13	1.82	-23.74	-5.61	-2.87	-0.70	1.05
50	10.29	5.96	193.54	21.09	0.13	2.22	-27.63	-9.89	-3.37	-1.24	1.13

Table 7: Results for different percentages of reduction in the light traveling cost.

age, and (ii) from stations with few or no deadheading options to nearby stations with more deadheading alternatives. This means that more owned locomotives can be conveniently made available at other stations through repositioning, reducing significantly the utilization of foreign units across the system. This, in turn, translates into a reduction in the total number of locomotives required to operate the train schedule.

6. Conclusion

In this paper we studied the LRP at the Canadian National Railway Company (CN), and proposed a large-scale integer linear programming formulation based on a two-layer time-space network representation of the problem. This graph lets us keep track of the maintenance status of specific locomotives over time, as well as managing the assignment of locomotives to trains based on their current maintenance status. Computational experiments performed on a set of real instances showed that our model is tractable and can be solved to optimality within reasonable computing times. In comparison to historical data, our methodology provides solutions that require fewer locomotives and less repositioning across the system. In addition, computational experiments showed that our model can be used to analyze alternative operational scenarios and support decision-making.

In practice, optimal solutions provided by our methodology represent only a guideline for real-time operations, which in turn must take into account several additional factors, such as train delays and locomotive breakdowns, all of which are subject to uncertainty. Providing more robust locomotive routes at the operational level is then essential to mitigate the impact of uncertain events on real-time operations. One way of achieving this is to explicitly consider one or several sources of uncertainty when modeling and solving the LRP. We will address this natural extension of the problem in subsequent research.

Acknowledgments

The authors gratefully acknowledge the collaboration of the Canadian National Railway Company (CN) and the funding received through the CN Chair in Optimization of Railway Operations, and a Collaborative Research and Development Grant from the Natural Sciences and Engineering Research Council of Canada (CRD-477938- 14). Computations were performed on the cluster *Béluga*, managed by Calcul Québec and Compute Canada, and funded by the Canada Foundation for Innovation (CFI), the ministére de l'Économie, de la science et de l'innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

Appendix A. Disaggregated Formulation

Consider the following additional notation.
Sets and parameters:

$\mathcal{V}^R$: Set of regular locomotives;

$\mathcal{V}^R_k$: Set of regular locomotives of type k;

λ^R_{vi} : Supply of regular locomotive v at source i;

η^R_{vl} : Flow of locomotive v on legacy arc $l \in A_{T^-} \cup A_{SH^-}$.

Decision Variables:

x_{vl} : 1 if regular locomotive v flows on arc l, 0 otherwise;

y_{vl} : 1 if critical locomotive v flows on arc l, 0 otherwise;

z_{kl} : Number of foreign (leased) locomotives of type k that flow on arc l;

u_{ki} : Number of additional locomotives of type k supplied by source i.

The disaggregated formulation for the LRP reads as follows:

$$\begin{aligned}\min \sum_{k\in\mathcal{K}} \sum_{l\in A^B_T \cup A_{DH} \cup A^B_L \cup A_G \cup A^B_Q} & c_{kl} \left(z_{kl} + \sum_{v\in\mathcal{V}^R_k} x_{vl} + \sum_{v\in\mathcal{V}^C_k} y_{vl} \right) \\ & + \sum_{v\in\mathcal{V}^C} \sum_{l\in A_{SH}(v)} c_{vl} y_{vl} + \sum_{v\in\mathcal{V}^C} \phi \left(1 - \sum_{l\in A_{SH}(v)} y_{vl} \right)\end{aligned} \tag{A.1}$$

subject to:

$$x_{vl} = \eta^R_{vl}, \qquad v \in \mathcal{V}^R, l \in A^B_{T^-} \cup A^B_{SH^-} \tag{A.2}$$

$$y_{vl} = \eta^C_{vl}, \qquad v \in \mathcal{V}^C, l \in A^-_T \tag{A.3}$$

$$\sum_{l\in O_v[i]} x_{vl} = \lambda^R_{vi}, \qquad v \in \mathcal{V}^R, i \in N^B_I \tag{A.4}$$

$$\sum_{i\in N_H^B}\sum_{l\in I_v[i]} x_{vl} = 1, \qquad v \in \mathcal{V}^R \tag{A.5}$$

$$\sum_{l\in I_v[i]} x_{vl} = \sum_{l\in O_v[i]} x_{vl}, \qquad v \in \mathcal{V}^R, i \in \{N_D^B \cup N_A^B \cup N_O^B \cup N_Q^B\} \tag{A.6}$$

$$\sum_{l\in O_v[i]} y_{vl} = \lambda_{vi}^C, \qquad v \in \mathcal{V}^C, i \in N_I \tag{A.7}$$

$$\sum_{i\in N_H}\sum_{l\in I_v[i]} y_{vl} = 1, \qquad v \in \mathcal{V}^C \tag{A.8}$$

$$\sum_{l\in I_v[i]} y_{vl} = \sum_{l\in O_v[i]} y_{vl}, \qquad v \in \mathcal{V}^C, i \in \{N_D \cup N_A \cup N_O \cup N_Q^B\} \tag{A.9}$$

$$\sum_{l\in O_k[i]} z_{kl} = u_{ki}, \qquad k \in \mathcal{K}, i \in N_I^B \tag{A.10}$$

$$\sum_{i\in N_H^B}\sum_{l\in I_k[i]} z_{kl} = \sum_{i\in N_I^B} u_{ki}, \qquad k \in \mathcal{K} \tag{A.11}$$

$$\sum_{l\in I_k[i]} z_{kl} = \sum_{l\in O_k[i]} z_{kl}, \qquad k \in \mathcal{K}, i \in \{N_D^B \cup N_A^B \cup N_O^B \cup N_Q^B\} \tag{A.12}$$

$$\sum_{v\in\mathcal{V}_k^R} x_{vl} + \sum_{v\in\mathcal{V}_k^C} y_{vl} + z_{kl} = \rho_{kl}, \qquad k \in \mathcal{K}, l \in A_T^B \tag{A.13}$$

$$\sum_{l\in A_{SH}(v)} y_{vl} \leq 1, \qquad v \in \mathcal{V}^C \tag{A.14}$$

$$\sum_{l\in A_{B2T}(v)} y_{vl} = 1 - \sum_{l\in A_{SH}^B(v)} y_{vl}, \qquad v \in \mathcal{V}^C \tag{A.15}$$

$$y_{vl'} = \sum_{l=(i,j)\in A_{SH}^T(v)} y_{vl}, \qquad v \in \mathcal{V}^C, l' = (j,h) \in A_{T2B}(v) \tag{A.16}$$

$$\sum_{v\in\mathcal{V}^C}\sum_{\substack{l=(j,h)\in A_{SH}(v):\\ p_j=s\\ t_j\leq t_i<t_h}} y_{vl} \leq C_s - \sum_{\substack{l=(j,h)\in A_{SH}^-:\\ p_h=s\\ t_j\leq t_i<t_h}}\sum_{v\in\mathcal{V}^R} \eta_{vl}^R, \qquad \begin{array}{l} s\in\mathcal{S}^{SH},\\ i\in N_{SH}^B:\\ p_i=s \end{array} \tag{A.17}$$

$$\sum_{k\in\mathcal{K}}\sum_{j\in E_{il}^B}\left(z_{kj} + \sum_{v\in\mathcal{V}_k^R} x_{vj} + \sum_{v\in\mathcal{V}_k^C} y_{vj}\right) + \sum_{v\in\mathcal{V}^C}\sum_{j\in E_{il}^T} y_{vj} \leq m_l^{DH}, \qquad \begin{array}{l} l\in A_T^B,\\ i=0,\ldots,n_l \end{array} \tag{A.18}$$

$$\sum_{k\in\mathcal{K}} z_{kl} + \sum_{v\in\mathcal{V}^R} x_{vl} + \sum_{v\in\mathcal{V}^C} y_{vl} \leq m^T, \qquad l \in A_L^B \tag{A.19}$$

$$\sum_{\substack{i \in N_F^B: \\ p_i = s}} \left(\sum_{l \in I_k[i]} z_{kl} + \sum_{v \in \mathcal{V}_k^R} \sum_{l \in I_v[i]} x_{vl} + \sum_{v \in \mathcal{V}_k^C} \sum_{l \in I_v[i]} y_{vl} \right) \geq m_{ks}, \qquad k \in \mathcal{K}, s \in \mathcal{S} \tag{A.20}$$

$$x_{vl} \in \{0,1\}, \qquad v \in \mathcal{V}^R, l \in \mathcal{A}^B \backslash A_{SH}^B \tag{A.21}$$

$$y_{vl} \in \{0,1\}, \qquad v \in \mathcal{V}^C, l \in \mathcal{A}^B \cup \mathcal{A}^T \cup A_{B2T} \cup A_{T2B} \tag{A.22}$$

$$z_{kl} \in \mathbb{Z}^+, \qquad k \in \mathcal{K}, l \in \mathcal{A}^B \backslash A_{SH}^B \tag{A.23}$$

$$u_{ki} \in \mathbb{Z}^+, \qquad k \in \mathcal{K}, i \in N_I^B. \tag{A.24}$$

The objective function (A.1) aims to minimize the total operational cost, which includes costs of pulling trains, deadheading, light traveling, idling at stations, enforcing train-to-train connections, locomotive maintenance, as well as penalties for not servicing critical locomotive by the end of the planning horizon. Constraints (A.2)–(A.3) impose initial conditions. Constraints (A.4)–(A.6) guarantee the proper flow of regular locomotives over the network, from sources to sinks. Likewise, Constraints (A.7)–(A.9) and (A.10)–(A.12) impose a proper flow of critical and leased locomotives, respectively.

Constraints (A.13) guarantee that each train is assigned the requested type and number of locomotives, considering regular, critical and foreign locomotives. Constraints (A.14) establish that each critical locomotive is moved to a shop at most once. Constraints (A.15) ensure that critical locomotives are moved to the overdue layer if they miss their maintenance deadline. Conversely, constraints (A.16) guarantee that overdue locomotives flow back to the service layer upon service in a shop. Shop capacity constraints are imposed by (A.17).

Constraints (A.18) limit the number of locomotives deadheading in a train, while (A.19) impose a limit on the number of locomotives on light traveling arcs. Constraints (A.20) enforce a minimum number of locomotives of each type at each station by the end of the planning horizon.

References

Ahuja, R., Magnanti, T., Orlin, J., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, New Jersey.

Ahuja, R.K., Cunha, C.B., Şahin, G., 2005a. Network Models in Railroad Planning and Scheduling. Tutorials on Operations Research (INFORMS). chapter 3. pp. 54–101. doi:10.1287/educ.1053.0013.

Ahuja, R.K., Liu, J., Orlin, J.B., Sharma, D., Shughart, L.A., 2005b. Solving Real-Life Locomotive-Scheduling Problems. Transportation Science 39, 503–517. doi:10.1287/trsc.1050.0115.

Benders, J.F., 1962. Partitioning Procedures for Solving Mixed-Variables Programming Problems. Numerische Mathematik 4, 238–252.

Bouzaiene-Ayari, B., Cheng, C., Das, S., Fiorillo, R., Powell, W.B., 2016. From Single Commodity to Multiattribute Models for Locomotive Optimization: A Comparison of Optimal Integer Programming and Approximate Dynamic Programming. Transportation Science 50, 366–389. doi:10.1287/trsc.2014.0536.

Cordeau, J.F., Soumis, F., Desrosiers, J., 2000. A Benders Decomposition Approach for the Locomotive and Car Assignment Problem. Transportation Science 34, 133–149. doi:10.1287/trsc.34.2.133.12308.

Cordeau, J.F., Soumis, F., Desrosiers, J., 2001. Simultaneous Assignment of Locomotives and Cars to Passenger Trains. Operations Research 49, 531–548. doi:10.1287/opre.49.4.531.11226.

Dantzig, G.B., Wolfe, P., 1960. Decomposition Principle for Linear Programs. Operations Research 8, 101–111. doi:10.1287/opre.8.1.101.

Ortiz-Astorquiza, C., Frejinger, E., Cordeau, J.F., 2019. The Locomotive Assignment Problem with Distributed Power At Canadian National Raylways. Technical Report CIRRELT 2019-55. Université de Montréal. Canada.

Piu, F., Prem Kumar, V., Bierlaire, M., Speranza, M., 2015. Introducing a Preliminary Consists Selection in the Locomotive Assignment Problem. Transportation Research Part E: Logistics and Transportation Review 82, 214–237. doi:10.1016/j.tre.2015.07.003.

Piu, F., Speranza, M.G., 2014. The Locomotive Assignment Problem: A Survey on Optimization Models. International Transactions in Operational Research 21, 327–352. doi:10.1111/itor.12062.

Powell, W.B., Bouzaiene-Ayari, B., Lawrence, C., Cheng, C., Das, S., Fiorillo, R., 2014. Locomotive Planning at Norfolk Southern: An Optimizing Simulator Using Approximate Dynamic Programming. Interfaces 44, 567–578. doi:10.1287/inte.2014.0741.

Rouillon, S., Desaulniers, G., Soumis, F., 2006. An Extended Branch-and-Bound Method for Locomotive Assignment. Transportation Research Part B: Methodological 40, 404–423. doi:https://doi.org/10.1016/j.trb.2005.05.005.

Scheffler, M., Neufeld, J.S., Hölscher, M., 2020. An MIP-based Heuristic Solution Approach for the Locomotive Assignment Problem Focussing on (Dis-)connecting Processes. Transportation Research Part B: Methodological 139, 64–80. doi:https://doi.org/10.1016/j.trb.2020.05.020.

Vaidyanathan, B., Ahuja, R.K., Liu, J., Shughart, L.A., 2008a. Real-life Locomotive Planning: New Formulations and Computational Results. Transportation Research Part B: Methodological 42, 147–168. doi:10.1016/j.trb.2007.06.003.

Vaidyanathan, B., Ahuja, R.K., Orlin, J.B., 2008b. The Locomotive Routing Problem. Transportation Science 42, 492–507. doi:10.1287/trsc.1080.0244.

Zhu, E., Crainic, T.G., Gendreau, M., 2014. Scheduled Service Network Design for Freight Rail Transportation. Operations Research 62, 383–400. doi:10.1287/opre.2013.1254.

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, 281–292. doi:10.1016/S0377-2217(96)00198-1.

Ziarati, K., Soumis, F., Desrosiers, J., Solomon, M.M., 1999. A Branch-First, Cut-Second Approach for Locomotive Assignment. Management Science 45, 1156–1168. doi:10.1287/mnsc.45.8.1156.