Les Cahiers du GERAD ISSN: 0711–2440

Locomotive assignment under consist busting and maintenance constraints

B. Jaumard, H. Tian,
P. Finnie

G–2014–54

August 2014

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Locomotive assignment under consist busting and maintenance constraints

Brigitte Jaumard[a]

Huaining Tian[b]

Peter Finnie[c]

[a] *GERAD & Computer Science and Software Engineering Department, Concordia University, Montréal (Québec) Canada, H3G 1M8*

[b] *Computer Science and Software Engineering Department, Concordia University, Montréal (Québec) Canada, H3G 1M8*

[c] *Canadian Pacific Railway, Calgary (Alberta) Canada, T2C 4X9*

bjaumard@cse.concordia.ca

August 2014

Les Cahiers du GERAD

G–2014–54

Abstract: We propose a new large scale optimization model, named TS_LAP, for the locomotive assignment problem (LAP), which relies on train strings or consist travel plans, i.e., sequences of trains assigned to the same consist. Although several algorithms have been developed for LAP, including exact mathematics models, approximate dynamic programming and heuristics, previously published optimization algorithms all suffer from scalability or accuracy issues. In addition, each of the previously proposed optimization models lacks part of the constraints that are necessary in real-world train/locomotive assignments, e.g., maintenance shop constraints or consist-busting avoidance. We propose a train string based LAP model, which includes those constraints and which can efficiently be solved using large scale optimization techniques, namely column generation techniques. Numerical results are conducted on the railway network infrastructure of Canada Pacific Railway, with up to 1,394 trains and 9 types of locomotives over a two-week time period. We investigate the impact of the size of the locomotive fleet on the consist busting and the deadheading operations.

Key Words: Locomotive assignment, large scale optimization, consist busting, column generation, maintenance.

Acknowledgments: B. Jaumard was supported by NSERC (Natural Sciences and Engineering Research Council of Canada) and by a Concordia University Research Chair (Tier I). H. Tian was supported by a FQRNT-NSERC-CPR fellowship.

1 Introduction

Freight train scheduling has already received a lot of attention, for its high energy efficiency and safety advantage, compared to other land and air transportation options. This paper focuses on one of the key elements of railway operation problems: the Locomotive Assignment Problem (LAP), which deals with the adequate size of a locomotive fleet and the assignment of locomotives to scheduled freight trains. The LAP aims to minimize the total number and/or cost of assigning locomotives to trains while all the technical and business constraints are satisfied, and consist busting is avoided as much as possible.

Several mathematical models and solution methodologies have been proposed for locomotive assignment. Most of them rely on multi-commodity network flow formulations and are either locomotive-based or consist-based. Both compact MILP (mixed integer linear program) or column generation formulations were investigated for exact solutions. Some other research studies investigated heuristics and approximate approaches to overcome the computational complexities. We next review the most comprehensive models or algorithms, see, e.g., Piu and Speranza [1] for a recent survey on LAP.

Ziarati et al. [2] focus on LAP in collaboration with CN (Canadian National), under a heterogenous fleet of locomotives. The model of Ziarati et al. [2] assumes that the source/destination stations of each train have the ability to attach/detach consists. The locomotive assignment schedule is assumed to be non-periodic over a 7-day horizon, and subject to time maintenance constraints. Consist busting is not taken into account. LAP is formulated as an integer multi-commodity time-space network flow problem, within a Column Generation (CG) framework (in which each pricing problem is associated with one commodity, i.e., one locomotive). Each node is associated with a railway station and a particular time, the arcs represent activities such as waiting/dwell periods, train travel between two stations (usually the origin and the destination) or train maintenance, and commodities are the locomotives. The authors solved data instances with up to 2,000 trains and 1,300 locomotives over a 7-day horizon, after decomposing the original problem into smaller overlapping problems, in order to handle the scalability issues. Rouillon et al. [3] improve the branch-and-price solution process of Ziarati et al. [2] with alternate branching methods.

Ahuja et al. [4] develop a MILP model for LAP, with a weekly cyclical train schedule, for CSX Transportation. The authors also formulate LAP as an integer multi-commodity flow problem. Unlike the one of Ziarati et al. [2], the model allows locomotive light traveling, but does not consider any maintenance constraint. Rather than using a mathematical programming solution scheme, the authors develop a neighborhood search algorithm/heuristic in order to solve large data instances. Another MILP model, with the same set of constraints, was recently proposed by Piu [5], but unfortunately no numerical results are presented, only a synthesis of the data sets and of the numerical results of the recent papers.

Most previously proposed LAP model are locomotive-based models and provide solutions with a high consist-busting rate, i.e., solutions in which consists of inbound trains are disassembled into stand alone locomotives and reassign to several outbound trains. However, some authors investigated some consist based models in which, instead of assigning locomotives, the model assigns consists to pull the scheduled trains subject to minimum power and other business constraints. Along those lines, Vaidyanathan et al. [6] develop a consist-based assignment model. However, the consist configurations are generated by a pre-processing algorithm, and therefore only a very limited number of consists are considered, and some constraints are missing, e.g., maintenance constraints.

Several heuristic solutions have been also developed in order to address the computational time and/or scalability issues of the deterministic optimization LAP models. Powell et al. [7] develop an approximate dynamic programming (ADP) approach, for short to long-term LAP, considering maintenance shop routing, transit time delays, dynamic schedule changes, and equipment failures. Godwin et al. [8] develop a heuristic based on Petri-Net model. [9] formulate LAP into the well-known vehicle routing problem with time windows (VRPTW), and solve it with a cluster-first, route-second approach. Indeed, they decompose the original multi-depot locomotive assignment into a series of single depot problems, each of which is independently solved by a hybrid genetic algorithm.

Previously published studies encompass most significant constraints in LAP, although there are no studies that include them all. In addition, whether assuming cyclic or non periodic train scheduling, most authors do not deal with the repositioning of the locomotives, a key issue in the search for minimum cost solution or at least workable solutions with respect to train demand and locomotive fleet size.

The paper is organized as follows. In Section 2, after a detailed statement of the locomotive assignment problem, we propose a new optimization model (TS_LAP) based on train strings (i.e., consist travel plans) for the locomotive assignment problem in the context of freight trains, e.g., the railway system of CPR - Canadian Pacific Railway. The resulting consist-based TS_LAP model (Section 3) uses a multi-commodity network flow formulation that includes maintenance constraints (i.e., maintenance due date and shop capacities) and avoidance of consist busting operations. We next present the solution scheme of the proposed TS_LAP model, which relies on a column generation decomposition algorithm that includes a train string generator, and consequently allows the inclusion of all possible consists throughout an *implicit* enumeration only. Numerical results are presented using several CPR locomotive and train data sets over the entire CPR network, from Vancouver to Montreal, including the U.S. railway component. Therein, among several analysis, we investigate the number of consist busting operations, the waiting/dwell times in between the arrival and departure times of the trains using the same consist, the best compromise between a larger locomotive fleet vs. more deadheading operations. Conclusions are drawn in the last section.

2 Problem statement

The Locomotive Assignment Problem (LAP) considers the allocation of a fleet of locomotives to pull a set of scheduled trains over a given time horizon. The objective is to minimize the operational locomotive cost, while satisfying the minimum power and/or tonnage train requirements as well as some other technical and business constraints: time maintenance constraints (assuming a calendar time in order to define the maintenance intervals), locomotive deadheading, locomotive repositioning for the next planning period, avoidance of consist busting operations. Indeed, the latter operations are a high time-consuming process and entails additional labor & operational cost and time requirements, while reducing the robustness of the train schedule.

The proposed optimization TS_LAP model, which will be detailed in the next section, builds an assignment of locomotives to scheduled/planned trains with respect to the above constraints.

Input of LAP includes: railway network, train parameters and schedules, and locomotive characteristics. The railway network can be described by a graph, where the nodes are associated with stations where consists can be de-assembled/re-assembled, and where maintenance operations can take place (in a restricted number of stations). Maintenance shop capacities are given, i.e., a limited number of locomotives can undergo maintenance in a given shop each day. Train demand is defined by the requested power, departure/arrival times and origin/destination stations of each scheduled train. Locomotive characteristics include their horsepower, their initial locations and types, and potentially their requested repositioning at the end of the planning period. If locomotive (re)location is not taken care, then potentially either more deadheading or more locomotives are needed in order to answer the requested train demand in the next planning period.

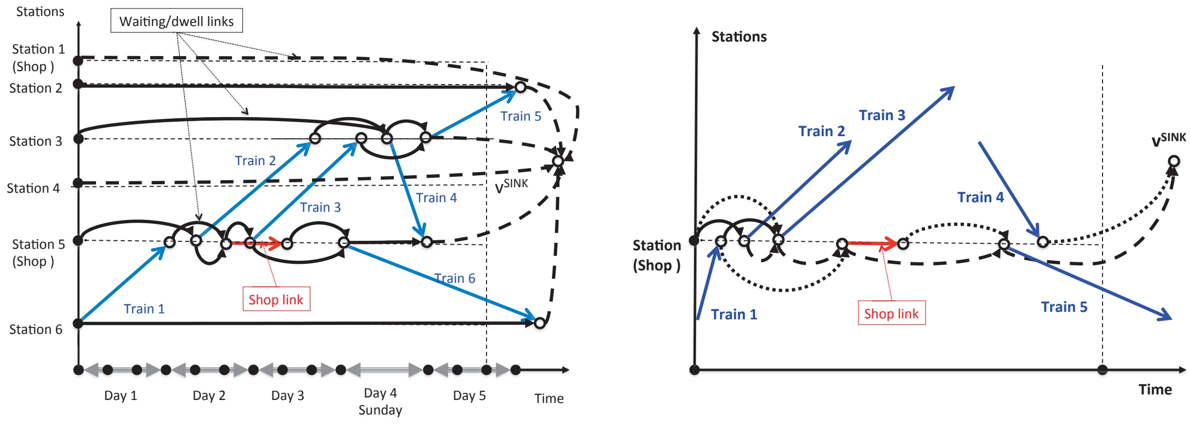
Output of LAP is the travel plan for each locomotive within a series of consist definitions and assignments, as well as deadheading (or light traveling) locomotives in order to satisfy the train traveling demand.

3 Locomotive assignment TS_LAP model

The proposed LAP (Locomotive Assignment Problem) model, called TS_LAP, relies on a multi-commodity network, which we next describe in Section 3.1. We next introduce the concept of train strings or consist travel plans (Section 3.2), the notations (Section 3.3), and then the TS_LAP model (Section 3.4).

3.1 Multi-commodity network

The multi-commodity network of the TS_LAP model is a time/space network, see Figure 1(a), where each node v is associated with two components: $\textsc{location}(v)$, which corresponds to a railway station location,



(a) Multi-commodity network

(b) Inbound and outbound waiting/dwell links

Figure 1: Definition of the various types of links of the multi-commodity network

and, TIME$(v) \in \mathrm{Z}^+$, which corresponds to the beginning or the end time of an activity, and which is expressed in minutes. The arcs represent activities such as waiting/dwell periods, train travel between two stations (usually the origin and the destination) or train maintenance, and commodities are the locomotives. It is sometimes convenient to view a multi-commodity network as a layered graph, where each layer is associated with one commodity, as some nodes and arcs may be specific to a particular commodity. We will assume here the same network for all types of locomotives.

When building the multi-commodity network for a given planning period, we take into account legacy trains, i.e., trains that departed in the previous planning period but which reach their destination in the current planning period, as well as trains departing in the planning period under study, which will reach their destination during the next planning period. For instance, in Figure 2, Train 1 departs before the beginning of the planning period under study.

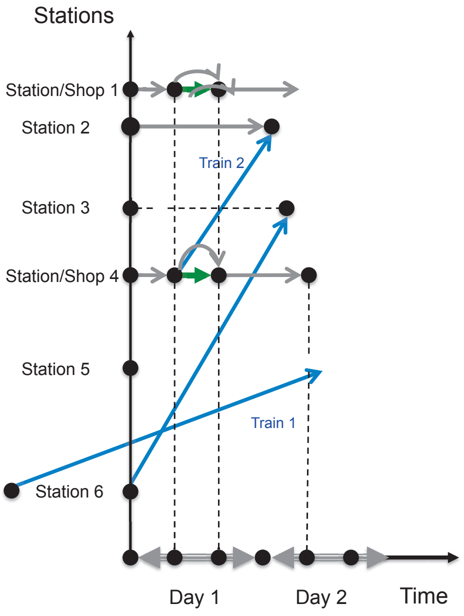


Figure 2: Definition of the V^{SRC} source node set

We now describe in detail the multi-commodity network $G = (V, L)$ associated with the overall set of locomotives, which we will use in the optimization TS_LAP model (Section 3.4). In order to simplify the exposure, we first omit the locomotive repositioning constraints, but come back on them in Section 5.3. Set V denotes the node set, indexed by v, where each v has a space and a time coordinate. Set L is the link set (indexed by ℓ) where

$$L = L^T \cup L^{\text{SHOP}} \cup L^W$$

where:

L^T is the set of links associated with trains.

$L^{\mathbf{shop}}$ is the set of links associated with the three month maintenance activity (regulatory requirement in North America), which takes place in a shop (located in the yard of a station). For $\ell \in L^{\text{SHOP}}$, we set its time origin to 8am, and its time destination to 5pm, so that it corresponds to one working day. In other words, $\ell = (\text{SHOP LOCATION}, 8\text{ am}) \rightarrow (\text{SHOP LOCATION}, 5\text{ pm})$.

$L^{\mathbf{w}}$ is the set of locomotive waiting/dwell links, i.e., associated with idle periods for a locomotive. For $\ell = (v, v') \in L^{\text{W}}$, time/space nodes v and v' are associated with the same station, i.e., $\text{LOCATION}(v) = \text{LOCATION}(v')$, for a locomotive remaining idle between $\text{TIME}(v)$ and $\text{TIME}(v')$. There are two critical threshold values: DWELL1_LOCO, the minimum amount of time in between two successive trains (one inbound, one outbound) with the same consist, and DWELL2_LOCO, the minimum amount of time in between two successive trains (one inbound, one outbound) sharing some, but not all locomotives.

We will not consider all possible waiting/dwell links, as they would be too numerous, but only the minimum number of them in order to ensure enforcing all locomotive assignment constraints.

To ensure the delay constraints between, e.g., two successive train departures (see constraints (3.4.3)), or for reassembling a consist, we partition the waiting/dwell links into the inbound and the outbound waiting/dwell links: $L^{\text{W}} = L^{\text{W\_IN}} \cup L^{\text{W\_OUT}}$ with $L^{\text{W\_IN}} \cap L^{\text{W\_OUT}} = \emptyset$.

An ***outbound waiting/dwell link*** ($\ell^{\text{W}} = (v, v') \in L^{\text{W\_OUT}}$) is such that: v is the source endpoint of a train or a shop link ; v' has a larger time component than v ($\text{TIME}(v') > \text{TIME}(v)$) and is the closest, but distinct, source of an outbound train or of a shop link at the same station as v. Potentially $v' = v^{\text{SINK}}$ if there is no more outbound train or shop link within the planning period, where v^{SINK} is a dummy node (see its definition below). In such a case, the link is called a sink link.

An ***inbound waiting/dwell link*** ($\ell^{\text{W}} = (v, v') \in L^{\text{W\_IN}}$) is such that: v is either such that $\text{TIME}(v) = 0$ (i.e., located in a station at the origin time of the planning period) or v is the destination endpoint of a train or a shop link ; v' has a larger time component than v and is the closest, but distinct, origin of an outbound train at the same station as v or origin of a shop link subject to the condition that: $(\text{TIME}(v') - \text{TIME}(v)) > \text{DWELL1\_LOCO}$. If $(\text{TIME}(v') - \text{TIME}(v)) < \text{DWELL2\_LOCO}$, then there is a second inbound waiting/dwell link originating at v towards v'' such that v'' is the closest, but distinct, origin of an outbound train at the same station as v or origin of a shop link subject to the condition that: $(\text{TIME}(v') - \text{TIME}(v)) > \text{DWELL2\_LOCO}$. Potentially, v' or v'' is equal to v^{SINK} if there is no more outbound train or shop link within the planning period, where v^{SINK} is a dummy node (see its definition below). In such a case, the link is called a sink link.

Inbound and outbound waiting/dwell links are illustrated in Figure 1(b) with dark arrows. Dash black arrows represent outbound waiting/dwell links, solid black arrows represent inbound waiting/dwell links with the DWELL1_LOCO time limit, and dotted black arrows represent inbound waiting/dwell links with the DWELL2_LOCO time limit.

The set of nodes contains all endpoints of the links of L. For each $\ell = (v, v') \in L^{\text{SHOP}}$, there is another link $\ell' = (v, v') \in L^W$ in order for a locomotive to bypass the maintenance step if not required or if the shop is full. Among the nodes, we identify some particular ones, i.e., the source nodes and a dummy sink node:

$V^{\mathbf{src}} = \bigcup_{k \in K} V_k^{\text{SRC}}$, where V_k^{SRC} is the set of nodes where some locomotives of type k (see Section 3.3 for the definitions of K, i.e., the locomotive type set) are initially positioned, i.e., available, in the planning period. Nodes of V_k^{SRC} are either nodes with a time component equal to the beginning of the planning period if a locomotive is idle in the corresponding station, or with a time component equal to the departure time of a legacy train, i.e., a train with a departure time prior to the beginning of the current planning period and an arrival time lying within the current planning period, see Figure 2. Therein, V^{SRC} contains 6 nodes (one associated with each station), 5 with a time index equal to the origin of the planning period, and one at an earlier time.

$v^{\mathbf{sink}}$, a sink node that is the collector node for all dummy sink links, in order to be able to enforce locomotive flow conservation at any node (except for the source and destination nodes) in the multi-commodity graph.

3.2 Train strings (i.e., consist travel plans)

We now introduce the concept of train strings. A train string is defined as a set of trains that use the same locomotive consist one train after the other one, without any consist busting, see Figure 3 for an illustration. A train can belong to at most one train string. If a train does not belong to any string, then it means there is a lack of available locomotives in order to pull it. Train strings (and shop links) are separated by waiting/dwell links, and must be spaced a minimum time (DWELL2_LOCO = 2 hours in our numerical experiments) in order to allow consists to be dismantled and then re-assembled. Within a train string, the time difference of two consecutive train links is spaced by a minimum time period, i.e., DWELL1_LOCO = 1 hour in our numerical experiments.

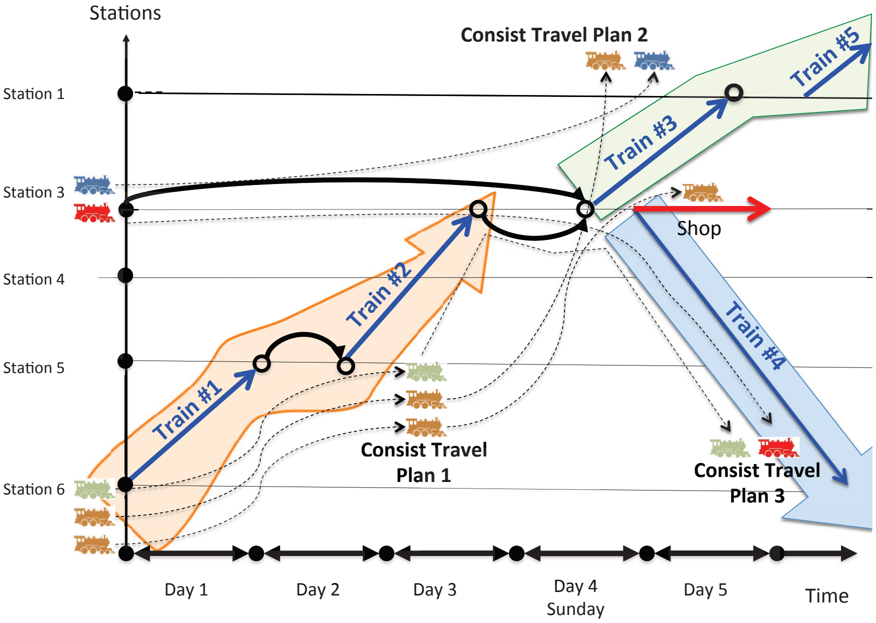


Figure 3: Some train strings, i.e., consist travel plans

3.3 Notations

3.3.1 Parameters

Trains and train strings. We denote by S the set of all possible train strings, indexed by s. We have:

$$S = \bigcup_{v \in V} S_v^+,$$

where S_v^+ denotes the set of train strings originating at v. Similarly, S_v^- denotes the set of train strings destined to v.

Moreover,

d^s_ℓ $= 1$ if train link $\ell \in L^T$ belongs to train string s, 0 otherwise

$\mathbf{day}^{\mathbf{src}}(t), \mathbf{day}^{\mathbf{dst}}(t)$**:** departure and arrival times of train t, expressed in days (assuming each day is represented by an integer value), using the start time of the planning period as a reference.

Locomotives. We denote by K the set of locomotive types, indexed by k. We decouple the set of locomotive types in order to distinguish the locomotives due for maintenance during the planning period from the other ones. Indeed, if a locomotive has its time maintenance due during the planning period, it is labeled as a critical locomotive that needs to undergo maintenance, i.e., to stop in a shop, before the maintenance day limit has expired. Maintenance can occur anytime between the beginning of the planning period and the ultimate maintenance due date. This implies that maintenance can take place earlier in order to take advantage of going through a station with a shop. If the planning period is quite large, one can easily add a constraint in order to avoid maintenance taking place too early (not done in this study). In terms of notations, when we need to distinguish regular (r) from critical (c) locomotives, we use the index k_c (resp. k_r), and decompose K into K_c and K_r.

Critical locomotives are relabelled as regular after completing their maintenance process at a shop. This relabelling will be taken care thanks to special flow conservation constraints, i.e., constraints (6), at the shop end nodes in the proposed TS_LAP model.

We introduce the following parameters:

n_k : number of available locomotives of type k throughout the railway system

n^s_k : number of locomotives of type k in train string $s \in S$

$n^{\mathbf{spare}}_{k,v}$: number of spare locomotives of type k at source node $v \in V^{\text{SRC}}$

$\text{CAP}(\ell^{\text{SHOP}})$: upper bound on the number of critical locomotives that can undergo maintenance in shop link $\ell^{\text{SHOP}} \in L^{\text{SHOP}}$ during one day.

m_k : total number of calendar days for the locomotives of critical type $k \in K_c$ since their last visit to a shop, until the start time of LAP scheduling period.

In order to alleviate the description of the model, we assume that all critical locomotives of a given type have the same value m_k. However, if it is not the case, it is easy to expand the model, by defining as many K_c sets as the number of different (or significantly different) m_k values.

Multi-commodity graph. In the multi-commodity graph $G = (V, L)$, we designate by $\omega(v)$ (resp. $\omega(V')$ with $V' \subseteq V$) the set of adjacent links to v (resp. to a node of V'). In addition, $\omega^+(v)$ (resp. $\omega^-(v)$) denotes the set of outgoing (resp. incoming) links of v. For a given link ℓ, $\delta^+(\ell)$ denotes the destination endpoint of ℓ, and $\delta^+(L'), L' \subseteq L$, denotes the set of destination endpoints of the links of L'. Similarly, $\delta^-(\ell)$ and $\delta^-(L'), L' \subseteq L$ denote the origin endpoint(s) of ℓ and of the links of L', respectively.

Set of incoming (resp. outgoing) waiting/dwell links of node v are denoted by $L_v^{\text{WAIT\_IN}}$ (resp. $L_v^{\text{WAIT\_OUT}}$). Similarly, set of incoming (resp. outgoing) shop links of node v are denoted by $L_v^{\text{SHOP\_IN}}$ (resp. $L_v^{\text{SHOP\_OUT}}$).

3.4 Optimization TS_LAP model

We next develop the LAP optimization model, called TS_LAP, that we propose for the locomotive assignment. In order to alleviate the presentation, we first describe it without legacy trains and locomotive repositioning constraints, see Section 5 for the required modifications of the TS_LAP model.

In addition, in order to guarantee that the model is always able to output a solution, we penalize the number of additional required locomotives (if any) in order to be able to pull all scheduled trains.

3.4.1 Variables

We use four sets of variables:

z_s = 1 if string s is selected, 0 otherwise.

$x^{\mathbf{need}}_{kv}$ = number of additional required locomotives of type k at source node $v \in V^{\text{SRC}}$ in order to be able to pull all trains with enough power, throughout an appropriate locomotive assignment to all trains. Trains with an additional locomotive corresponds to trains without proper power with the existing locomotive fleet. Then, either the railway company needs to rent a locomotive or to delay the train in order to pull the trains associated with so-called additional locomotives.

$x^{\mathbf{loco}}_{k\ell}$ = number of locomotives of type k on link ℓ.

Note that:

$$x^{\text{LOCO}}_{k\ell^{\text{W}}} = x^{\text{LOCO}}_{k_r\ell^{\text{W}}} + x^{\text{LOCO}}_{k_c\ell^{\text{W}}}, \quad x^{\text{LOCO}}_{k_r\ell^{\text{SHOP}}} = 0, \quad x^{\text{LOCO}}_{k\ell^{\text{SHOP}}} = x^{\text{LOCO}}_{k_c\ell^{\text{SHOP}}}.$$

3.4.2 Objective

The primary objective is to minimize both the number of consist busting and the size of the locomotive fleet. While the minimization of those two numbers seem to go in opposite directions, the maintenance constraints force to withdraw locomotives from the tracks for a short period, hence creating some unavoidable consist busting. Moreover, there is a compromise between the locomotive fleet size and the number of deadheading locomotives: a reduced locomotive fleet entails more deadheading and consist busting operations. We will investigate this point in Section 6.6 and then propose an objective aiming to the minimization of: *(i)* the number of locomotive occurrences; *(ii)* the number of total locomotives in operation during the planning period. This second objective component is motivated by the observation that the initial locomotive positioning is critical in order to find a feasible locomotive assignment (i.e., which can properly pull all scheduled trains), and is investigated in the numerical experiments in Section 6.3.

$$\min \sum_{s \in S} \sum_{k \in K} n^s_k \, z_s + \sum_{\ell \in \omega^-(v^{\text{SINK}})} \sum_{k \in K} \text{PENAL}_k \cdot x^{\text{LOCO}}_{k\ell}, \tag{1}$$

where PENAL_k is a penalty factor in order to give a higher priority to the second objective component.

3.4.3 Constraints

Spare locomotive constraints. For each locomotive type, for each station location $\text{LOCATION}(v)$, the overall number of locomotives required for trains departing form $\text{LOCATION}(v)$ cannot exceed the number of spare ones in $\text{LOCATION}(v)$ at the outset of the planning period. In order to avoid facing infeasible locomotive assignments, we introduce variables x^{NEED}_{kv} that compute the number of missing locomotives (or the number of additional required locomotives) in order to accommodate all scheduled trains. It is enough to introduce them for the regular locomotives. The number of trains to $\text{LOCATION}(v)$ is estimated by the number of outbound waiting/dwell links ending in v. Ensuring that we do not use more locomotives than the number of spare ones (up to the correcting term x^{NEED}_{kv}) can then be expressed as follows:

$$\sum_{s \in S^+_v} n^s_k \, z_s + \sum_{\ell \in \omega^+(v) \cap L^{\text{WAIT}}} x^{\text{LOCO}}_{k\ell^{\text{W}}} - x^{\text{NEED}}_{kv} \leq n^{\text{SPARE}}_{k,v} \qquad k \in K_r, v \in V^{\text{SRC}} \tag{2}$$

$$\sum_{s \in S^+_v} n^s_k \, z_s + \sum_{\ell \in \omega^+(v) \cap L^{\text{WAIT}}} x^{\text{LOCO}}_{k\ell^{\text{W}}} \leq n^{\text{SPARE}}_{k,v} \qquad k \in K_c, v \in V^{\text{SRC}} \tag{3}$$

$$\sum_{\ell \in \omega^-(v^{\text{SINK}})} x^{\text{LOCO}}_{k\ell} \leq n_k \qquad k \in K. \tag{4}$$

Constraints (4) guarantee that, even if we allow the usage of additional locomotives, the overall number of used locomotives can not exceed the size of the locomotive fleet, i.e., the maximum number of locomotives of each type. Constraints (4) also serve the purpose of deadheading locomotives whenever it is possible, before either renting locomotives in order to pull all scheduled trains, or delaying a train due to a lack of power.

Flow conservation constraints. For a given node (except for the source and sink nodes), the number of outbound locomotives must be equal to the number of inbound ones for each type of locomotives. We first consider the nodes which are not the destination endpoints of a shop link (so that no critical locomotive need to be relabelled regular while going through such a node):

$$\sum_{s \in S_v^+} n_k^s \, z_s + \sum_{\ell \in \omega^+(v) \,\cap\, L^{\text{WAIT}}} x_{k\ell}^{\text{LOCO}} = \sum_{s \in S_v^-} n_k^s \, z_s + \sum_{\ell \in \omega^-(v) \,\cap\, L^{\text{WAIT}}} x_{k\ell}^{\text{LOCO}}$$
$$v \in V \setminus \left(V^{\text{SRC}} \cup \{v^{\text{SINK}}\} \cup \delta^+(L^{\text{SHOP}})\right), k \in K_r \cup K_c. \quad (5)$$

At the destination endpoint of a shop link, i.e., after the completion of maintenance operations, critical locomotives that went under maintenance need to be relabelled as non critical ones, of the same type (see an illustration in Figure 4):

$$\sum_{s \in S_v^+} n_{k_r}^s \, z_s + \sum_{\ell \in L_v^{\text{WAIT\_OUT}}} x_{k_r\ell}^{\text{LOCO}} = \sum_{s \in S_v^-} n_{k_r}^s \, z_s + \sum_{\ell \in L_v^{\text{SHOP\_IN}}} x_{k_c\ell}^{\text{LOCO}} + \sum_{\ell \in L_v^{\text{WAIT\_IN}}} x_{k_r\ell}^{\text{LOCO}}$$
$$v \in \delta^+(L^{\text{SHOP}}), k = \{k_r, k_c\} \in K \quad (6)$$

$$\sum_{s \in S_v^+} n_k^s z_s + \sum_{\ell \in L_v^{\text{WAIT\_OUT}}} x_{k\ell}^{\text{LOCO}} = \sum_{s \in S_v^-} n_k^s z_s + \sum_{\ell \in L_v^{\text{WAIT\_IN}}} x_{k\ell}^{\text{LOCO}}$$
$$v \in \delta^+(L^{\text{SHOP}}), k \in K_c. \quad (7)$$

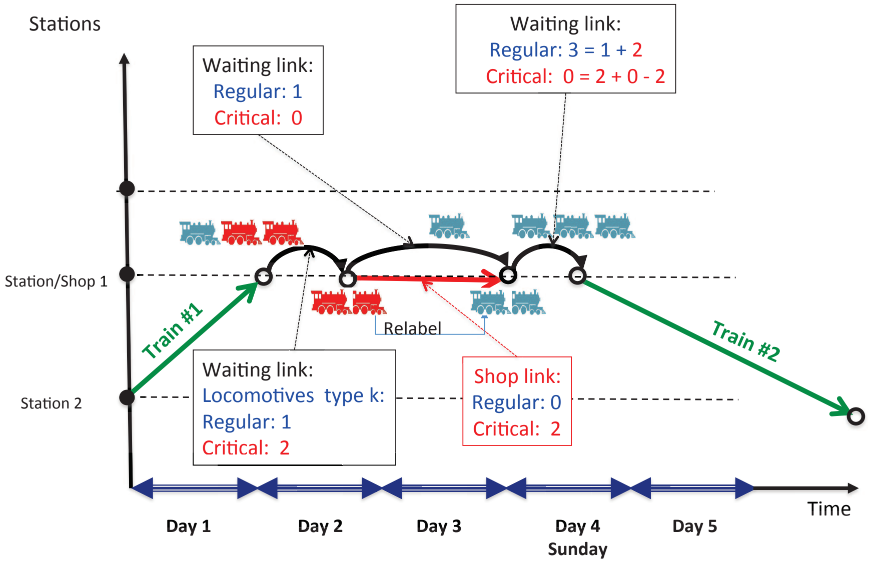


Figure 4: Critical locomotive change their status to regular after going through a maintenance shop (red locomotives are critical, while the blue ones are regular)

Train disjoint string constraints. Each train should belong to **exactly** one train string in the locomotive assignment:

$$\sum_{s \in S} d_\ell^s \cdot z_s = 1 \qquad \ell \in L^T. \quad (8)$$

Consist busting constraints. We must always allow for some time between any two consecutive train strings that share some locomotives. For that reason,

$$\sum_{k \in K} x^{\text{LOCO}}_{k\ell^{\text{W}}} = 0 \qquad \ell^{\text{W}} \in \ell^{\text{W\_IN}} \setminus \omega^{+}(V^{\text{SRC}}) : \textsc{time}(\ell^{\text{W}}) < \textsc{dwell2\_loco}. \tag{9}$$

As a particular case, a node v cannot be the source and the destination of two train strings with non disjoint locomotive consists.

As a consequence, at any node in the multi-commodity network, the locomotives that are assigned to the first train in any outbound train string departing from v, must be taken from either the set of locomotives that are idle for more than DWELL2_LOCO ago, or from a shop. Those constraints can be expressed as follows:

$$\sum_{s \in S^{+}_{v}} n^{s}_{k}\, z_s \leq \sum_{\ell^{w} \in L^{\text{WAIT\_IN}}_{v}} x^{\text{LOCO}}_{k\ell} \qquad v \in V \setminus \left(V^{\text{SRC}} \cup v^{\text{SINK}} \cup \delta^{+}(L^{\text{SHOP}})\right), \quad k \in K_r \cup K_c \tag{10}$$

$$\sum_{s \in S^{+}_{v}} n^{s}_{k_r}\, z_s \leq x^{\text{LOCO}}_{k_c\ell^{\text{SHOP}}} + \sum_{\ell \in L^{\text{WAIT\_IN}}_{v}} x^{\text{LOCO}}_{k_r\ell} \qquad v : v \in \delta^{+}(\ell^{\text{SHOP}}), \quad \ell^{\text{SHOP}} \in L^{\text{SHOP}}, k = \{k_r, k_c\} \in K \tag{11}$$

$$\sum_{s \in S^{+}_{v}} n^{s}_{k} z_s \leq \sum_{\ell \in L^{\text{WAIT\_IN}}} x^{\text{LOCO}}_{k\ell} \qquad v : v \in \delta^{+}(\ell^{\text{SHOP}}), \quad \ell^{\text{SHOP}} \in L^{\text{SHOP}}, k \in K_c. \tag{12}$$

Constraints (10) take care of a node v that is not the destination endpoint of a shop link. In such a case, available locomotives at v have been idle for more than DWELL2_LOCO, thanks to constraints (9) and the definition of the waiting/dwell links. We do not account for the locomotives made available by any train string that would destined v.

Constraints (11) and (12) take care of a node v that is the endpoint of a shop link. Constraints (11) deal with regular locomotives, while constraints (12) deal with critical locomotives. Regular locomotives can be made available either on the waiting/dwell links assuming they have been available for more than DWELL2_LOCO, which is the case thanks to constraints (9), or at the exit of shops. Critical locomotives can be made available only from the waiting/dwell links.

Shop capacity constraints. For each shop station, the number of critical locomotives that are allowed at the same time for maintenance is limited and should not exceed the shop capacity:

$$\sum_{k_c \in K} x^{\text{LOCO}}_{k_c\ell^{\text{SHOP}}} \leq \textsc{cap}(\ell^{\text{SHOP}}) \qquad \ell^{\text{SHOP}} \in L^{\text{SHOP}}. \tag{13}$$

Deadheading. The identification of the deadheading locomotives is done in a post-processing phase, where we identify the locomotives which are not needed for pulling a specific train, but have been assigned to it in order to be transferred to another station where they are needed for pulling other trains.

4 Solution of the TS_LAP model

To solve the TS_LAP model developed in the previous section, we need to use column generation techniques in order to avoid the exhaustive enumeration of the train strings, and limit their generation to the improving ones. We therefore use a solution scheme that is summarized in the flowchart of Figure 5, and which is described in next section.

4.1 Solution process

We establish a first model with an initial set of train strings, called Restricted Master Problem (RMP), while the Master Problem corresponds to the model described in Section 3.4. The Restricted Master Problem is solved alternately with the generation of an augmenting train string, i.e., a train string such that, if added to the current RMP, will lead to an enhanced RMP with a smaller optimal value than the one of the previous RMP. Indeed, the generation of such a train string corresponds to the solution of the so-called pricing problem, see, e.g., Chvatal [10]. This latter problem either generates an improving train string, i.e., a string whose addition leads to a new restricted master problem with a smaller optimal value, or concludes that the current solution of the RMP is indeed the optimal solution of the linear relaxation of the Master Problem. It then remains to generate an integer solution, which can be easily done using an ILP solver or an iterative rounding off procedure on the last RMP (see Section 4.2), and then evaluate the accuracy of the resulting integer solution, see next section. We next discuss how to set the pricing problem in Section 4.3.

4.2 Column generation and integer solution

As illustrated in the flowchart of Figure 5, the solution process consists of two phases: the optimal solution of the linear relaxation of the master problem with the column generation technique, and then the derivation of an integer solution thanks to a branch-and-bound algorithm (CPLEX MILP solver) applied to the last solved Restricted Master Problem, i.e., the one with all the z_s variables generated until we reach the optimal solution of the linear relaxation of (1)–(13). While we are not guaranteed to generate a strict optimal solution, it allows the generation of an ε-optimal solution, where

$$\varepsilon = \frac{\tilde{z}_{\text{ILP}} - z^{\star}_{\text{LP}}}{z^{\star}_{\text{LP}}},$$

with $z^{\star}_{\text{LP}}$ being the optimal value of the linear relaxation of the Master Problem as described in (1)–(13), and $\tilde{z}_{\text{ILP}}$ being the optimal ILP value of the last solved Restricted Master Problem (but not necessarily the optimal integer value of the TS_LAP model).

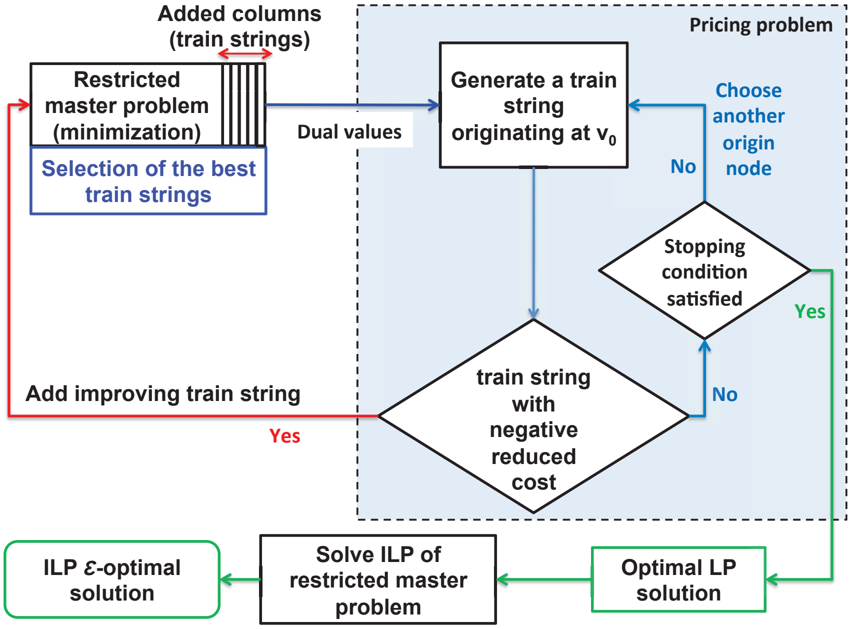

Added columns
(train strings)
Pricing problem
Restricted master problem (minimization)
Selection of the best train strings
Dual values
Generate a train string originating at v_0
Choose another origin node
No
Stopping condition satisfied
Yes
train string with negative reduced cost
No
Add improving train string
Yes
ILP ε-optimal solution
Solve ILP of restricted master problem
Optimal LP solution

Figure 5: Flowchart: column generation process

Note that the linear relaxation of the master problem is not solved optimally by embedding all possible train strings, but only a very small subset of them. At each iteration of the column generation algorithm, one more train string is added, for a given node $v \in V$, from where the first train of the train strings comes. Indeed, the column generation algorithm consists in a set of rounds, where, in each round, the algorithm goes through each source node v of a train link, and check whether a train string originating at v and with a negative reduced cost can be generated thanks to the pricing problem. If, during a round, the algorithm fails to find at least one train string with a negative reduced cost, the algorithm has reached the optimal solution of the linear programming relaxation, $z^{\star}_{\text{LP}}$.

4.3 Pricing problem: Generation of an augmenting train string

Train strings are generated by the so-called pricing problem for a given origin node v_{O} for the train string under construction. In order to generate an exact solution, the column generation algorithm must consider all possible origin nodes, i.e., all nodes associated with the departure time of a train.

4.3.1 Variables

A train string starts with a train link, and ends at the endpoint of (another or the same) train link.

We use the following set of variables in order to build train string s (index s is omitted in order to alleviate the notations):

$\mathbf{dst}_v$ = 1 if train string s under construction ends at node $v \in \delta^+(L^T)$ (destination endpoint of a train link), 0 otherwise.

x_ℓ = 1 if link $\ell \in L^T$ belongs to the path supporting the train string under construction, 0 otherwise. Note that this variable is in one-to-one correspondence with parameter d^s_ℓ of the master problem.

n_k = number of locomotives of type k in the train string under construction.

4.3.2 Objective: Reduced cost of the z_s variables

The objective of the pricing problem is the so-called reduced cost (if not familiar with linear programming concepts, the reader is referred to, e.g., Chvatal [10]) of the z_s variables. The s index is omitted in this in order to alleviate the notations.:

$$\begin{aligned}\overline{\text{COST}} = \sum_{k \in K} n_k \quad &+ \quad \left[\sum_{k \in K_r} u^{(2)}_{kv_0} n_k + \sum_{k \in K_c} u^{(3)}_{kv_0} n_k\right] \\ &- \sum_{k \in K_r \cup k_c} \left(u^{(5)}_{kv_0} n_k - \sum_{v \in V \setminus (V^{\text{SRC}} \cup \{v^{\text{SINK}}\} \cup \delta^+(L^{\text{SHOP}}))} u^{(5)}_{kv} \, n_k \text{DST}_v \right) \\ - \sum_{k \in K_r} \left(u^{(6)}_{kv_0} \cdot n_k - \sum_{v \in \delta^+(L^{\text{SHOP}})} u^{(6)}_{kv} \, n_k \text{DST}_v \right) &- \sum_{k \in K_c} \left(u^{(7)}_{kv_0} \cdot n_k - \sum_{v \in \delta^+(L^{\text{SHOP}})} u^{(7)}_{kv} \, n_k \text{DST}_v \right) \\ + \sum_{k \in K_r \cup K_c} \; \sum_{v \in V \setminus (V^{\text{SRC}} \cup \{v^{\text{SINK}}\} \cup \delta^+(L^{\text{SHOP}}))} & u^{(10)}_{kv} \, n_k \text{DST}_v + \sum_{v \in \delta^+(L^{\text{SHOP}})} u^{(11)}_{kv} \, n_k \text{DST}_v \\ &+ \sum_{v \in \delta^+(L^{\text{SHOP}})} u^{(12)}_{kv} \, n_k \text{DST}_v - \sum_{\ell \in L^T} u^{(8)}_t \, x_\ell. \quad (14)\end{aligned}$$

Note that the reduced cost objective (14) is nonlinear, because of the quadratic terms $n_k \cdot \text{DST}_v$, which can be linearized using the product variables y_{kv}:

$$y_{k,v} \leq n_k \qquad v \in V, k \in K \qquad (15)$$

$$y_{k,v} \leq M \cdot \text{DST}_v \qquad v \in V, k \in K \qquad (16)$$

$$y_{k,v} \geq n_k + M \cdot (\text{DST}_v - 1) \qquad v \in V, k \in K. \qquad (17)$$

Note that constraints (16) can replaced by the unique constraint:

$$\sum_{k \in K} y_{k,v} \leq M \cdot \text{DST}_v \qquad v \in V \qquad (18)$$

in order to reduce the number of constraints (although potentially deteriorating the linear relaxation lower bound).

Note that when the pricing problem is solved for an origin node v_{v_0} that does not belong to V_{SRC}, the term between square bracket in the above expression should be omitted. Lastly, the terms with summation over v in (14) can be reduced to the summation over the node subset $v \in \delta^+(L^T)$.

4.3.3 Constraints of the train string generator

A unique destination node. Ensure that there is exactly one destination node for the train string under construction, and it has to be the endpoint of a train link.

$$\sum_{v \in \delta^+(L^T)} \text{DST}_v = 1. \tag{19}$$

Definition of the train string. The next set of constraints corresponds to flow conservation constraints in order to define a path in the time-space graph, i.e., the set of trains belonging to the train string under construction:

$$\sum_{\ell \in \omega^+(v)} x_\ell - \sum_{\ell \in \omega^-(v)} x_\ell = -\text{DST}_v \qquad v \in \delta^+(L^T) \tag{20}$$

$$\sum_{\ell \in \omega^+(v)} x_\ell - \sum_{\ell \in \omega^-(v)} x_\ell = 0 \qquad v \in V \setminus \delta^+(L^T) \cup \{v_\text{O}\} \tag{21}$$

$$\sum_{\ell \in \omega^+(v_\text{O}) \cup L^T} x_\ell = 1 \tag{22}$$

$$\sum_{\ell \in \omega^-(v_\text{O})} x_\ell = 0. \tag{23}$$

No empty train string. Ensure that the train string under construction is not empty.

$$\sum_{\ell \in L^T} x_\ell \geq 1. \tag{24}$$

No overdue critical locomotives. Critical locomotives need to stop by a shop before the maintenance due date. The next set of constraints ensure that the train string contains no critical locomotives (definitions in page 4) that are overdue for calendar maintenance:

$$n_k \leq M \cdot (1 - x_\ell) \qquad \ell \in L^T, k \in K_c : \text{TIMEDST}(t) + m_{k_c} \geq \text{MAINT\_TIME}, \tag{25}$$

where MAINT_TIME was set to 90 days in the numerical experiments.

Note: TIMEDST(t) counts the number of days from the start time of LAP plan period, to the arrival time of train t.

Power constraints. Ensure that each train included in the train string under construction has enough power.

$$\sum_{k \in K_r} \text{HP}_k \cdot n_k + \sum_{k \in K_c} \text{HP}_k \cdot n_k \geq \max_{\ell (\equiv t) \in L^T} \{x_\ell \cdot \text{HP}_t\}. \tag{26}$$

Bounds on the consist size. Constraints (27) ensure upper and lower bounds for the consist size:

$$\text{consist\_size}_{\text{min}} \leq \sum_{k \in K_r \cup K_c} n_k \leq \text{consist\_size}_{\text{max}}. \tag{27}$$

No back-to-back train. The next set of constraints prevent back-to-back trains:

$$\sum_{\ell \in \omega^+(v_o) \cup L^T} x_\ell + \sum_{\ell \in \omega^-(v_o) \cup L^T} x_\ell \leq 1 \qquad v \in V. \tag{28}$$

5 Legacy trains, deadheading and locomotive repositioning

5.1 Legacy trains

In order to take into account the legacy trains, we need to modify the multi-commodity network as follows. Firstly, we add the train links associated with those trains such that their departure time is before the beginning of the planning period, but their arrival time falls during the planning period. We add the waiting/dwell links resulting from those additional train links. Last, we add the origin of those legacy train links to the set of source nodes. All the constraints remain the same on the modified multi-commodity network.

5.2 Deadheading

In the numerical experiments, we investigated the impact of the deadheading locomotives with respect to the size of the locomotive fleet. For that purpose, we consider a different objective, i.e., minimize the discrepancy between the required and the assigned power to the scheduled trains:

$$\min \quad \sum_{s \in S} \sum_{t \in s} z_s \cdot (\text{HP}_s - \text{HP}_t). \tag{29}$$

We also added a constraint in order to limit the size of the locomotive fleet:

$$\sum_{v \in V^{\text{SRC}}} \sum_{k \in K} x_{kv}^{\text{NEED}} \leq \text{UB}^{\text{need}}. \tag{30}$$

Following the change of objective (from (1) to (29)), the expression of the reduced cost in the pricing problem needs to be updated accordingly. We leave it to the reader.

Next, we modify the power constraints (26) as follows:

$$\sum_{k \in K_r} \text{HP}_k \cdot n_k + \sum_{k \in K_c} \text{HP}_k \cdot n_k = x_\ell(\text{HP}_t + \text{DEAD}_\ell) \qquad \ell \in L^T, \tag{31}$$

where $\text{DEAD}_\ell \in \mathbb{R}^+$ represents the deadheading power for each train link $\ell^t \in L^T$. Again, the quadratic term $x_\ell \text{DEAD}_\ell$ can be easily linearized using the product variables α_ℓ:

$$\alpha_\ell \leq Dead_\ell \qquad \ell \in L^T \tag{32}$$

$$\alpha_\ell \leq M \cdot x_\ell \qquad \ell \in L^T \tag{33}$$

$$\alpha_\ell \geq Dead_\ell + M \cdot (x_\ell - 1) \qquad \ell \in L^T. \tag{34}$$

5.3 Locomotive re-position constraints

If we do not worry of the locomotive repositioning for the next planning period, based on the expected set of scheduled trains, we may need much more locomotives than needed, as shown in the numerical results of Section 6.3. We can therefore set some demand constraints, in terms of the minimum number of locomotives of a given type required in the various stations, at the end of the planning period.

In order to do so, we need to modify the definition of the multi commodity flow network as follows, see Figure 6 for an illustration. Sets L^{TRAINS} and L^{SHOP} remain defined as before. For the waiting/dwell links, only the destination endpoints v of the train links such that $\text{TIME}(v) >$ end time of the planning period are

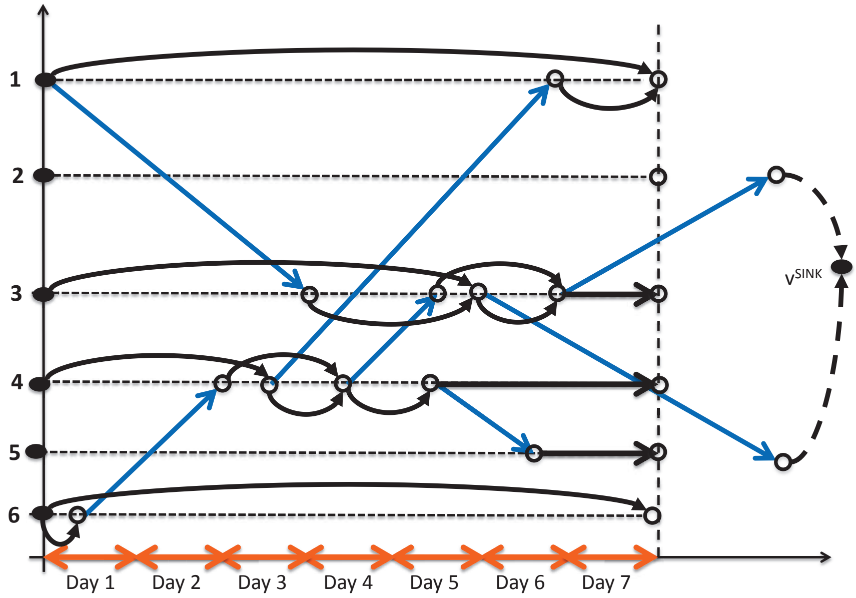


Figure 6: Revised multi-commodity network

linked to the sink node. Definitions of the inbound and outbound links remain the same, except that they cannot reach v^{SINK}. Lastly, there is a waiting/dwell link towards each node v such that $\text{TIME}(v) =$ end time of the planning period, and the origin of those links are the nodes with the same station location, and the largest time component that is different from the end time of the planning period.

At the end of the LAP planning period, the locomotive demand constraints are expressed as follows. For all $v : \text{TIME}(v) =$ end time of the planning period,

$$\sum_{s \in S_v^-} n_k^s z_k + \sum_{\ell^{\text{W}} \in \omega^-(v)} x_{k\ell^{\text{W}}}^{\text{LOCO}} \geq \text{DEMAND}_{\text{LOCATION}}^{k}(v) \qquad k \in K_r. \tag{35}$$

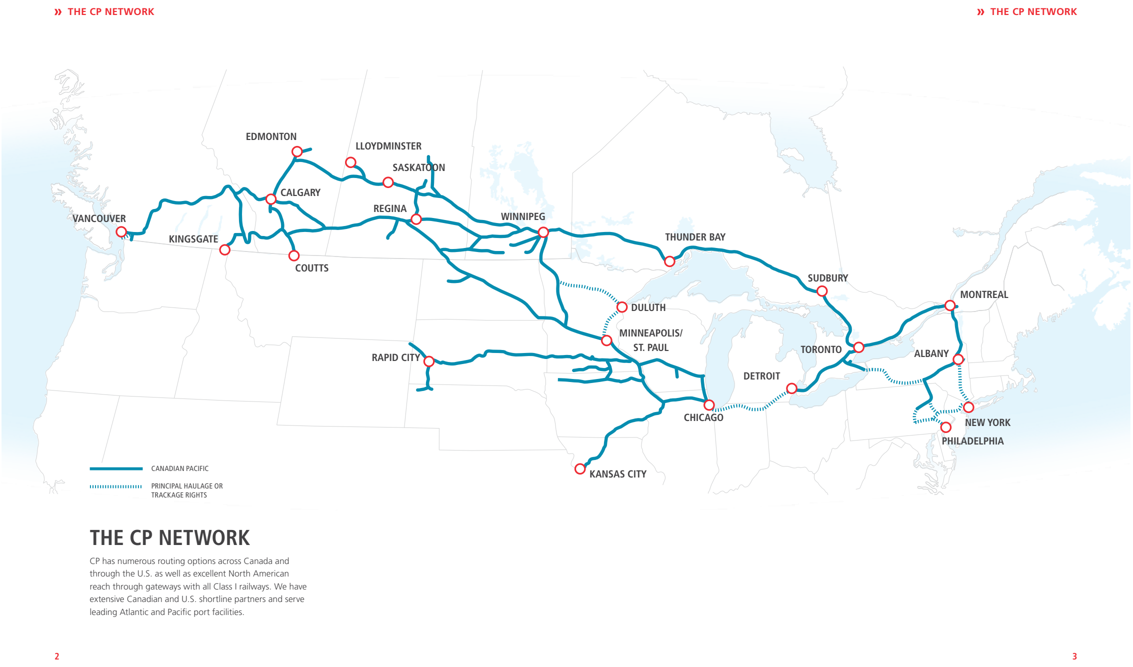
6 Numerical results

We implemented the model TS_LAP and solve it on various Canadian Pacific Railway (CPR) data sets. After analyzing the accuracy of the solutions (Section 6.2), we convey several analysis with respect to the (re)positioning of the locomotives (Section 6.3), the in between train waiting/dwell times in train strings (Section 6.4), the power assignment (Section 6.5) and the number of deadheading locomotives vs. the size of the locomotive fleet (Section 6.6).

6.1 Data instances

We use the overall CPR railway network between Vancouver and Montreal, spanning both Canada and parts of the United States, as shown in Figure 7), together with the location and the capacity of the shops. Data include a set of scheduled trains with their departure times and stations, train arrival times and stations, and horse-power requirements.

Data also include a set of locomotives, with the number of locomotives for each locomotive type. We use a set of 9 different types of locomotives, limiting our experiments to the most used locomotives in the CPR fleet of locomotives, as described in Table 1. As requested by the mathematical model, the number of types was decoupled in order to distinguish the critical (about 20% of the overall number of locomotives) from the regular (non critical) locomotives.



THE CP NETWORK

CP has numerous routing options across Canada and through the U.S. as well as excellent North American reach through gateways with all Class I railways. We have extensive Canadian and U.S. shortline partners and serve leading Atlantic and Pacific port facilities.

Figure 7: CPR railway network (taken from [11])

Table 1: Locomotives types & quantity

Model	Horsepower	units
GP38	3,000	196
GP40	3,000	89
GP40-2	3,000	89
SD40-2	3,000	307
SD60	3,800	232
SD90/4300	4,300	89
ES44AC	4,360	89
AC4400CW	4,400	1,143
AC4400CW-L	4,390	89
TOTAL	N/A	2,323

We defined 13 data sets with an increasing number of scheduled trains, from 103 to 1,394 trains. The largest data set corresponds the typical number of CPR scheduled trains over a time period of 2 weeks. Programs were run on *briaree* cluster (each node has two Intel Westmere EP X5650 six-core processors with 24 or 48 GB of memory), using the computing resources of [12].

As the data of the initial locomotive location was not available to us, we use a heuristic in order to generate an initial geographical distribution of the locomotive fleet over the stations in the railway network. This last step is critical as it plays a crucial role in order to make sure that a given locomotive fleet is well distributed so as to be able to pull all scheduled trains.

6.2 Accuracy and efficiency of the solutions

We now analyze the characteristics of the solutions of the TS_LAP model. For each data set (103 up to 1,394 trains), results are summarized in Table 2.

The second column shows the number of train strings (columns) generated by the TS_LAP model, i.e., the pricing problem, excluding those in the original input. The third column provides the number of train

Table 2: Characteristics of the TS_LAP solutions (heuristic locomotive geographical distribution)

	# Columns		# Locomotives			
# Trains	Generated	Selected (# Train strings)	Current fleet	Additional	ε (%)	Computing times (hh:mm:ss)
103	58	100	274	49	0.77	00:01:14
207	112	194	524	84	0.26	00:02:59
307	139	277	759	120	0.62	00:05:17
421	191	356	778	138	0.39	00:10:10
525	67	498	961	145	0.29	00:04:13
624	81	580	1,092	137	0.16	00:06:02
743	94	698	1,171	107	0.41	00:08:45
842	104	781	1,270	144	0.64	00:11:52
936	114	857	1,355	128	0.76	00:15:27
1,037	127	948	1,445	166	0.60	00:25:52
1,137	140	1,047	1,544	154	1.01	00:28:11
1,234	149	1,112	1,626	137	0.80	00:36:01
1,394	497	1,186	1,652	173	0.71	04:15:42

strings selected in the final solution for the ILP model. We therefore conclude that very few columns need to be generated before reaching the optimal solution of the linear relaxation of the TS_LAP model. The number of train strings with one more than one train is limited due to the two week planning period. Indeed, there are 55 of them, and the train composition is displayed in Figure 9, see Section 6.4 for an explanations of the data that are represented.

The fourth column shows the number of locomotives that are used. Indeed, due to the initial locomotive distribution, only a fraction of the locomotives can be used (about 70%). As a consequence, as indicated in the fifth column, there is a need for a large number of additional locomotives in order to be able to pull all trains.

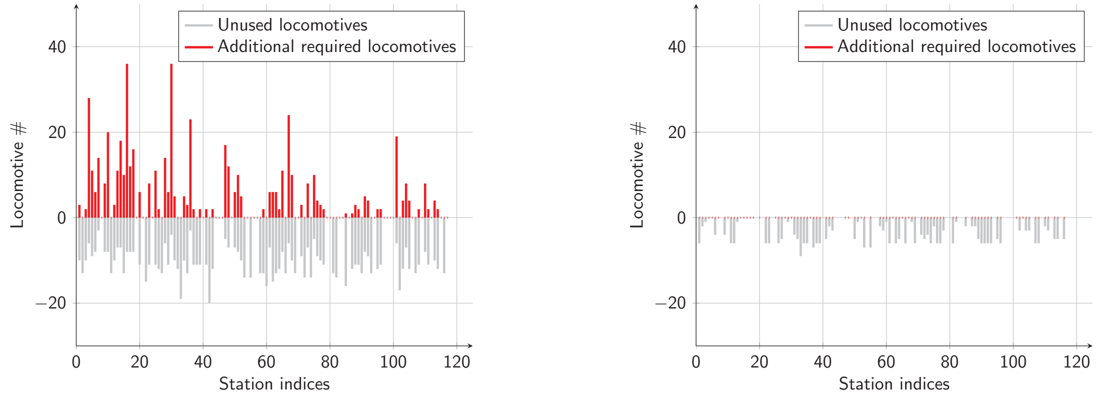
Accuracy (ε), i.e., the relative gap between the optimal value of the linear relaxation ($z^{\star}_{\text{LP}}$) and the value of the integer solution ($\tilde{z}_{\text{ILP}}$) is given in the sixth column. The relative gap is never than 1.1%, meaning that all ILP solutions are very close to the optimal ILP value. Computational times are given in the last column: they are significantly smaller than those reported in the literature taking into account the size of the largest data set, and that the TS_LAP model contains more constraints than all previously proposed LAP models.

We expect that, with the addition of a better heuristic in order to generate an initial set of consists and locomotive assignments, we can significantly reduce those computing times, and therefore solve even larger instances or the same instances over a longer planning period.

6.3 Initial locomotive positioning

Looking at the results in Table 2, i.e., a large number of additional requested locomotives (173 additional locomotives for the data set with 1,394 scheduled trains) in order to pull all scheduled trains while only 70 % of spare locomotives are used, we can see that the initial distribution plays a critical role. As plotted in Figure 8(a) for the data set with 1,394 scheduled trains, about 30% of spare locomotives are unused, and the number of unused locomotives varies with the stations.

We then re-initialize the input data, using an initial distribution of the locomotives based on the solution output by the TS_LAP. The new solutions are described in Table 3. We then observe than no additional locomotive are required, while still only using the same set of spare locomotives as in the previous experiment. In addition, the number of train strings remains the same, with a significantly smaller number of generated train strings in the column generation process. The accuracy of the solutions slightly degrades, but the computational times are significantly smaller. Figure 8(b) shows the new distribution of unused locomotives.



(a) Heuristic initial locomotive distribution

(b) Optimized initial locomotive distribution

Figure 8: Locomotive initial position for 1,394 train scenario

Table 3: Characteristics of the TS_LAP solutions (optimized locomotive geographical distribution)

	# Columns		# Locomotives			
# Trains	Generated	Selected (# Train strings)	Current fleet	Additional	ε (%)	Computing times (hh:mm:ss)
936	87	857	1,355	0	2.31	00:15:02
1,037	93	948	1,445	0	2.55	00:20:08
1,137	110	1,047	1,544	0	3.79	00:27:17
1,234	127	1,112	1,626	0	6.72	00:40:30
1,394	121	1,186	1,652	0	0.12	00:37:12

6.4 In between train waiting/dwell times in train strings

In this section, we analyze the in between train waiting/dwell times, for each train string. Figure 9 displays the train time configuration in each train string containing more than one train, for the data set with 1,394 trains. The horizontal axis represents the index of the train strings (the one-train strings are dismissed), the vertical axis represents the calendar time, with a one day unit, for the overall planning period. Each vertical

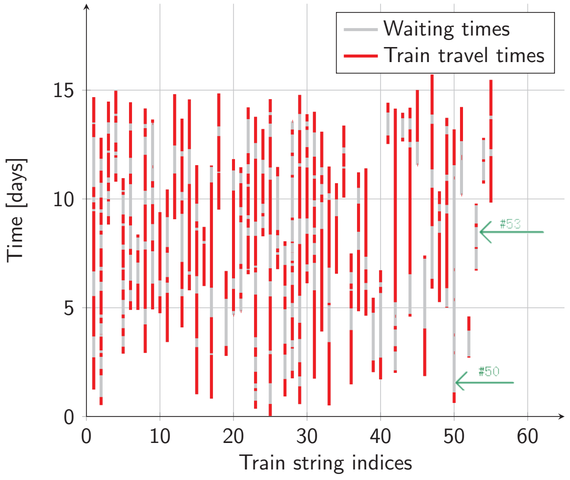


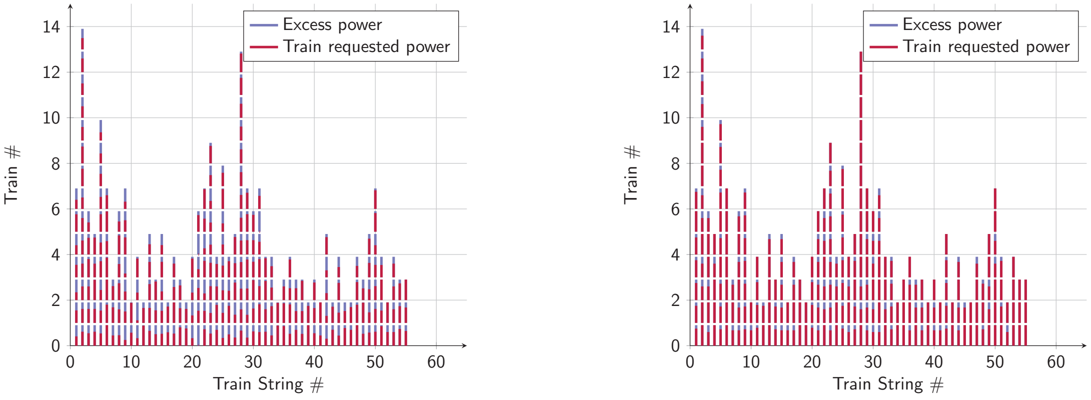
Figure 9: Waiting/dwell in between trains of train strings for the data set with 1,394 trains

bar represents a train string from the departure time of its first train to the arrival time of its last train. The red segments of the bar represent the train travelling times, and the grey segments are the in-between train waiting/dwell times.

We can see that, for most train strings, the in-between train waiting/dwell times are very close to the DWELL_LOCO time. This reflects the efficiency of the LAP solution. There are few exceptions, e.g., train strings #50 and #53, which indeed contains local trains.

6.5 Assigned vs. requested train power

We now propose to discuss the potentially over powered trains and the number of deadheading locomotives in each train string. Figure 10 shows the excess power, or wasted power, (expressed in percentage) for each train in each train string (the one-train strings have been dismissed), for the data set with 1,394 trains. In Figure 10(a), we analyze the requested vs. assigned power for each train. The horizontal axis represents the indices of the train strings, the vertical axis represents the number of trains in each train string. Each vertical bar represents a train string, and each unit is associated with a train. In each unit, the red part represents the percentage of requested power over the overall assigned power. Consequently, the blue part is the percentage of excess or wasted power.



(a) Excess power before discounting the deadheading locomotives

(b) Excess power after discounting the deadheading locomotives

Figure 10: Excess power for the data set with 1,394 trains

However, the excess power can be over-estimated, due to the presence of un-identified deadheading locomotives. For instance, consider train t that is requiring 10,000HP, and is assigned a consist of 4 locomotives of 4,400HP each. If we assume all locomotives to be active, there is an excess of 7,600HP. However, if we take into account that only 3 locomotives are active, the requested train power is still met (13,200 provide HP vs. 10,000 requested HP), and the excess power drops to 3,200. So, in Figure 10(b), we adjust the excess power estimation after identifying the deadheading locomotives (they may differ from one train to the next in a given train string).

The resulting excess power that is shown in Figure 10(b) is reduced. While it might be safe to have a small power excess, it also shows that some train may be able to pull a slightly larger number of cars.

6.6 Locomotive fleet size vs. consist busting and deadheading

Railway companies are interested in identifying the minimum number of required locomotives in order to pull all their scheduled trains. However, there is a compromise between the minimum number of locomotives and the number of deadheading locomotives. In the next set of experiments, we investigate how the number of deadheading locomotives is increasing when the locomotive fleet is reduced.

Results are shown in Figure 11. The horizontal axis is associated with the number of locomotives: the origin corresponds to the current size of the locomotive fleet, and then we increase the number of locomotives. The left vertical (red) axis represents the number of deadheading locomotives, while the right vertical (blue) axis represents the number of train strings. We observe that, as we increase the number of locomotives, the number of deadheading locomotives reduces until it reaches a threshold value (around 150). On the other hand, as the number of locomotives is increasing, the number of train strings increases as well until it reaches the number of scheduled trains. This is due to the objective of the TS_LAP model, which aims to minimize the discrepancy between the assigned and the required power: as there are 9 types of locomotives, and quite different required power values, the best consist varies from one train to the next.

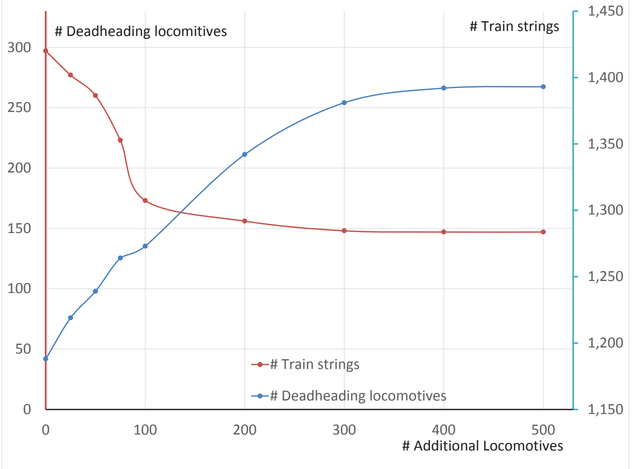


Figure 11: Deadheading vs. size of the locomotive fleet

7 Conclusions

We proposed a new mathematical model for optimizing the planning of locomotive assignment to a set of scheduled trains, subject to all meaningful constraints including the maintenance ones. The resulting model allows the *exact* solution of larger data instances than reported so far in the literature in a reasonable amount of time. In the future, we plan to enhance the scalability through the addition of heuristics in order generate an initial solution and to solve the pricing problems. Moreover, we plan to add the mileage maintenance constraints that are often used in combination with the time maintenance constraints.

References

[1] F. Piu, M. Speranza, The locomotive assignment problem: a survey on optimization models, International Transactions in Operational Research 21 (3) (2013) 327–352.

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

[3] S. Rouillon, G. Desaulniers, F. Soumis, An extended branch-and-bound method for locomotive assignment, Transportation Research. Part B, Methodological 40 (5) (2006) 404–423.

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

[5] F. Piu, A mixed integer programming approach to the locomotive assignment problem, in: Swiss Transport Research Conference (STRC), 2011, pp. 1–42.

[6] B. Vaidyanathan, R. Ahuja, J. Liu, L. Shughart, Real-life locomotive planning: new formulations and computational results, Transportation Research Part B 42 (2008) 147–168.

[7] W. Powell, B. Bouzaiene-Ayari, C. Cheng, R. Fiorillo, S. Das, C. Lawrence, Strategic, tactical and real - time planning of locomotives at Norfolk Southern using approximate dynamic programming, in: Joint Rail Conference (JRC), 2012, pp. 491–500.

[8] T. Godwin, R. Gopalan, T. Narendran, Tactical locomotive fleet sizing for freight train operations, Transportation Research Part E: Logistics and Transportation Review 44 (3) (2008) 440–454.

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

[10] V. Chvatal, Linear Programming, Freeman, 1983.

[11] Canadian Pacific Railway Limited, The CP book, Tech. rep., Canadian Pacific, Calgary, Alberta, Canada (2012). URL `http://www.cpr.ca/en/invest-in-cp/investor-book/Documents/2012-cp-investor-book.pdf`

[12] Last visit on August 10, 2014 (2014). [link]. URL `http://www.calculquebec.ca/en/`