Multi-Column Generation Model for the Locomotive Assignment Problem

Brigitte Jaumard*[1] and Huaining Tian[2]

1 **Computer Science and Software Engineering, Concordia University, Montreal (Qc) Canada H3G 1M8**
`bjaumard@cse.concordia.ca`

2 **Computer Science and Software Engineering, Concordia University, Montreal (Qc) Canada H3G 1M8**
`h_tia@encs.concordia.ca`

Abstract

We propose a new decomposition model and a multi-column generation algorithm for solving the Locomotive Assignment Problem (LAP). The decomposition scheme relies on consist configurations, where each configuration is made of a set of trains pulled by the same set of locomotives. We use the concept of conflict graphs in order to reduce the number of trains to be considered in each consist configuration generator problem: this contributes to significantly reduce the fraction of the computational times spent in generating new potential consists. In addition, we define a column generation problem for each set of variables, leading to a multi-column generation process, with different types of columns.

Numerical results, with different numbers of locomotives, are presented on adapted data sets coming from Canada Pacific Railway (CPR). They show that the newly proposed algorithm is able to solve exactly realistic data instances for a timeline spanning up to 6 weeks, in very reasonable computational times.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases Railway optimization, Locomotive assignment, Column Generation

Digital Object Identifier 10.4230/OASIcs.ATMOS.2016.6

1 Introduction

Rail transport is a very energy efficient means of freight transport. Compared to road transport using trucks, it consumes substantially less energy. Consequently, in many countries, governments are developing policies in order to encourage the use of trains for freight transport. At the same time, at least in North America, freight railways have increasingly shifted toward using longer, heavier trains to transport goods over the past 10 years, in order to not only improve the efficiency of the rails by reducing the number of trains required to transport goods, but also to reduce the crews needed and the fuel used to move their shipments. One consequence is that more locomotives need to be assigned to a single train, and then locomotive assignment becomes a critical problem in view of locomotive costs, and the objective of maintaining the smallest possible locomotive fleet.

Management of a locomotive fleet includes the assignment of proper locomotives to each train in schedule, satisfying the horsepower requirements, remotely relocating locomotives

* B. Jaumard was supported by NSERC (Natural Sciences and Engineering Research Council of Canada) and by a Concordia University Research Chair (Tier I).

16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'16).
Editors: Marc Goerigk and Renato Werneck; Article No. 6; pp. 6:1–6:13
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

for the trains, and making sure to obey to the locomotive maintenance rules. In this paper, we focus solely on the locomotive assignment problem. The set of locomotives that is used to pull a given train is called a **consist**. Note that today, some railway industry use the so-called distributed power trains, in which the locomotives are interspersed throughout the full length of the train, cutting down on the in-train forces and making the near-boundless vehicle easier to control. Beyond the distributed power system, a time-consuming process is called **consist busting**. It corresponds to disassembling the consist of an inbound train into stand alone locomotives and reassigning them to several outbound trains. It requires additional labor, induces operational cost and time. It also reduces the robustness of the train schedules because it allows an outbound train to get locomotives from multiple inbound trains. If any of the inbound trains is delayed, the outbound train has to be delayed as well. So consist busting should be avoided as much as possible.

We focus on the optimization of locomotive assignment problem (LAP), which aims to optimize the locomotive fleet size to satisfy the horsepower requests of scheduled trains and the other technical and business constraints. The objective of LAP is not only to minimize the total number of locomotives in operation, but to view the locomotive management as a whole, i.e., with the integration of the minimization of locomotive number and operational/maintenance costs.

There are many solution methodologies proposed for locomotive assignment, including exact mathematical models and heuristics. In this paper, we concentrate on the former part, and the latter part can be found in the survey of Piu *et al.* [13].

Ziarati *et al.* [19] focus on LAP with heterogeneous consists, i.e., made of different types of locomotives. In addition, the locomotive assignment also includes the need to perform some maintenance shopping and outpost process. In order to get a feasible solution in a reasonable computational time, Ziarati *et al.* decompose the original 1-week problem into several sub-problems which have overlapping day between adjacent ones. Rouillon *et al.* [14] improve the solution algorithm of Ziarati *et al.* [19] with different branching methods and search strategies to develop a branch-and-price algorithm for LAP of a freight railway on operational level. Ahuja *et al.* [1] develop a MILP for LAP of CSX Transportation for a cyclic weekly train schedule. However, the maintenance process, i.e., routing back to shop site for critical locomotive is not considered. The authors develop a neighborhood search algorithm/heuristic to improve the performance for large scale data instances, with no information on the accuracy of the output solutions. Ahuja's model neither considers locomotive maintenance nor consist busting issue. To avoid the issues of the model of Ahuja *et al.* [2] (e.g., scalability and consist busting issues), Vaidyanathan *et al.* [18] focus on a consist based assignment model, which assigns pre-configured consists to pull the scheduled trains with respect to the minimum power and other business constraints. Their consist based formulation uses a data set with 382/388 trains, 6 locomotive types, 87 stations, and 3 up to 17 types of consists in the test scenarios, without considering the maintance/shopping constraints for locomotives.

There are also some model for similar problems. Cordeau *et al.* [6] propose a exact model based on the Benders decomposition approach, for the locomotive and car assignment in passenger transportation. Fügenschuh *et al.* [8] propose an ILP model for the locomotive and car cycle scheduling problem with time window, which allow the train delay within given time window. Cacchiani *et al.* [3] focus on the train unit assignment problem in passenger transportation. A type of train unit includes a set of passenger cars with a supported locomotive. It is self-contained and may fulfill one or part of a scheduled trip. They propose two integer linear programming formulations, one with linear programming (LP) relaxation based heuristic, the other with Lagrangian relaxation based heuristic. None of these three models consider maintenance constraint, neither for consist busting issue.

In our previous papers (Jaumard *et al.* [10] & [11]), we proposed a consist travel plan(previously called train string) based optimization model, which includes all those constraints including maintenance, and consist busting constraints. The model can be efficiently solved using large scale optimization techniques, namely column generation techniques, to optimize the locomotive requirements and the operations including consist busting and the deadheading. The resulting model can solve LAP with up to 1,394 scheduled trains and 9 types of locomotives over a two-week time period, over the railway network infrastructure of Canada Pacific Railway. The computational time of the largest test scenario took more than 2 days.

In this paper, we propose to enhance the scalability of our previous work, throughout a multi-column generation strategy.

Other authors have explored various strategies at different stages of column generation algorithms, which can accelerate the computational time or the convergence rate. Firstly, in the pre-processing stage, there are heuristics that can reduce the initial size of original problem, e.g., for network flow problem, to eliminate arcs for the initial network (e.g., Mingozzi *et al.* [12]), to initialize with a good-enough solution (e.g., Sadykov *et al.* [16]), to separate a large scale problem to smaller parts in time or space horizon, and merge them after (e.g., Desaulniers *et al.* [7]). Secondly, in the sub-problem stage, Chen *et al.* [4] use some problem-specific knowledge to generate a column-pool a priori for the subproblem, and allow selections of solutions from the pool. In column generation practice, some schemes allow a subproblem to return multiple columns with negative reduced cost. Goffin *et al.* [9] observe that the non-correlated columns selection increases the performance in the analytic center cutting plane method. At the master problem level, Surapholchai *et al.* [17] develop Eligen-algorithm that applies column elimination which removes columns with positive reduced cost from the matrix. Saddoune *et al.* [15] use dynamic constraint aggregation to reduce number of constraints and reintroduce them as needed are two general strategies. Sadykov *et al.* [16] use a diversified diving heuristic to get feasible and good integer solution.

This paper is organized as follows. In Section 2, we generally describe LAP. Section 3 gives the details of LAP model. In Section 4, the solution scheme for the model is presented with two enhanced schemes/algorithms. In Section 5, we analyze the numerical results.

2 Statement of LAP

The locomotive assignment problem (LAP) is to minimize the total number and/or cost of assigning locomotives on existing trains while all the technical and business constraints are satisfied. A locomotive fleet usually contains different types of locomotives, e.g., SD60 and AC4400CW, each with its own parameters. We do not distinguish the locomotives of same type, except for their maintenance status (regular or critical). A critical locomotive, i.e., a locomotive due to maintenance, must stop at a shop for maintenance operations during the given scheduled time period. A consist travel planis defined as a set of trains that use the same locomotive consist one train after the other one, without any consist busting.

2.0.0.1 Multi Commodity Network

The present study focuses on reducing the size and time consumption of pricing problem for the decomposition of LAP. As in our previous study [10], we convert LAP to a multi-commodity network problem. The multi commodity network is a time/space network, see

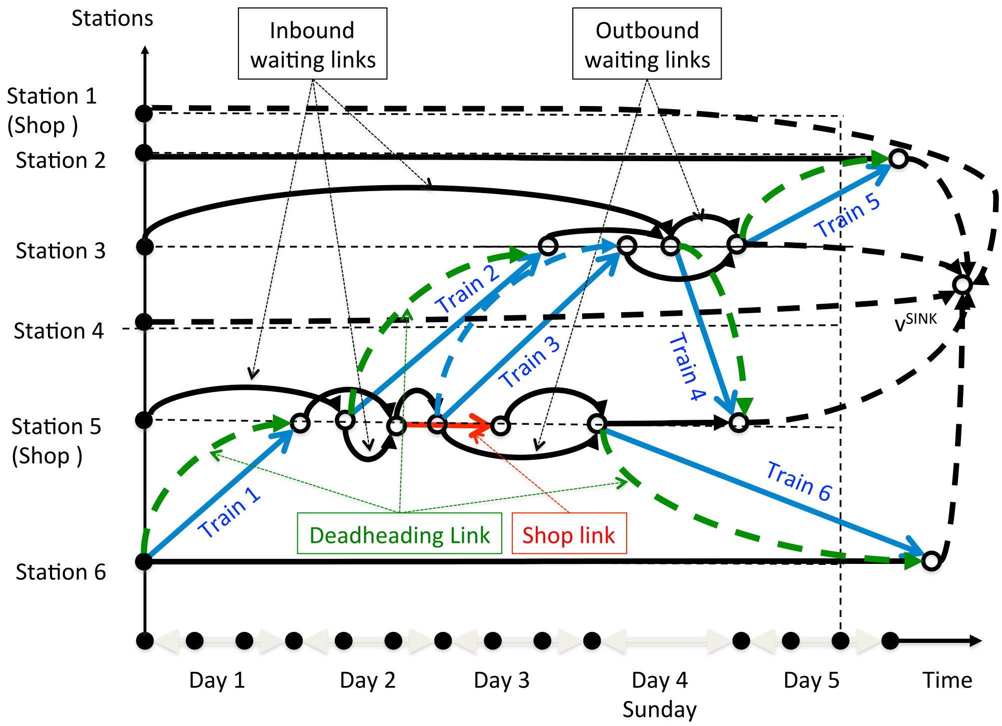


Figure 1 Multi commodity network.

Figure 1, where each node v is associated with station location, and time. The arcs represent activities such as waiting periods, train travel between two stations (usually the origin and the destination) or train maintenance, and commodities are the locomotives.

We now describe in detail the generic multi-commodity network $G = (V, L)$ associated with the overall set of locomotives. V denotes the set of nodes, indexed by v, where each v has a space and a time coordinate. L is the set (indexed by ℓ) where $L = L^T \cup L^{\text{SHOP}} \cup L^W \cup L^D$ which represents train links, maintenance shop links, waiting links and deadheading links respectively.

Among the nodes, we identify the so-called source and destination nodes as follows: V^{SRC}: indexed by v^{SRC}, as the set of nodes where some locomotives are first available in the planning period. v^{SINK}: dummy destination node, where all destination arcs converge. See the links represented by the long dash lines in Figure 1 for an illustration.

3 LAP Model

3.1 Notations

S is the set of consist travel plans, where a consist travel plan$s \in S$ defines a sequence of trains led by the same locomotive consist. Note that $S = \bigcup\limits_{v \in V} S_v^+$, where S_v^+ (resp. S_v^-) denotes the set of consist travel plansoriginating at (resp. destined to) v.

K denotes the set of locomotives, indexed by k, which represents a certain locomotive.

Each locomotive k is characterized by different parameters: the horsepower HP_k, and the subtype of regular (indexed by k_r) and critical (k_c).

Moreover, we use the following additional parameters to characterize the generated consist travel plans:
$n_k^s \in \{0, 1\}$. n_k^s is equal to 1 if locomotive k belongs to consist travel plan $s \in S$, 0 otherwise.
$n_{k,v}^{\text{SPARE}} \in \{0, 1\}$. $n_{k,v}^{\text{SPARE}}$ is equal to 1 if there is a spare locomotive k in starting node $v \in V^{\text{SRC}}$, 0 otherwise.
$d_\ell^s \in \{0, 1\}$. d_ℓ^s is equal to 1 if train link $\ell \in L^T$ belongs to consist travel plan$s$, 0 otherwise. Note that d_ℓ^s is not a decision variable, but an attribute of consist travel plan$s$.
$n_{k,v}^{\text{SPARE}} \in Z_0^+$. It is equal to the number of spare locomotives of type k in source node $v \in V^{\text{SRC}}$.

Lastly, we have the following last general parameters:
$\text{CAP}(\ell^{\text{SHOP}}) \in Z_0^+$. It defines an upper bound on the number of critical locomotives that can be maintained in shop link $\ell^{\text{SHOP}} \in L^{\text{SHOP}}$.
$\text{TimeSrc}(t), \text{TimeDst}(t) \in Z_0^+$. They define the start and end times (in days) of train t, counted from the start time of LAP scheduling period.

3.2 Variables.

We use three sets of variables:
$z_s \in \{0, 1\}$: equals to 1 if ctp s is selected, 0 otherwise.
$x_{kv}^{\text{NEED}} \in Z_0^+$: number of additional required locomotives of type k at source node $v \in V^{\text{SRC}}$ in order to be able to assign adequate locomotives to all trains.
$x_{k\ell}^{\text{LOCO}} \in Z_0^+$: number of locomotive of type k going through waiting link $\ell \in L^W \cup L^D \cup L^{\text{SHOP}}$.

3.3 Objective

We next develop the LAP optimization model we propose for the locomotive assignment. In order to alleviate the presentation, we describe it without the legacy trains.

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 avoidable consist busting. Moreover, if the locomotive fleet is too small, depending of the train schedule, there might be a lack of locomotives in order to be able to move all the trains. We therefore propose the following objective with the minimization of: *(i)* the number of consist travel plans; *(ii)* the number of additional locomotives, which reflects the number of trains that can not be assigned enough power; *(iii)* the number of total locomotives in operation, *(iv)* the number of deadheading activities.

$$\begin{aligned}\min \quad & \sum_{\ell \in \omega^-(v^{\text{SINK}})} \sum_{k \in K} \text{PENAL}_k \cdot x_{k\ell}^{\text{LOCO}} + \sum_{\ell \in L^D} \sum_{k \in K} \text{PENAL}_k \cdot x_{k\ell}^{\text{LOCO}} \\ & + \sum_{v \in V^{\text{SRC}}} \sum_{k \in K} \text{PENAL}_k \, x_{kv}^{\text{NEED}} + \sum_{s \in S} \sum_{k \in K} n_k^s \, z_s \end{aligned} \tag{1}$$

3.4 Constraints

The set of constraints can be written as follows.

$$\sum_{s \in S_v^+} n_k^s\, z_s + \sum_{\ell^w \in \omega^+(v)} x_{kL^{\text{W}}}^{\text{LOCO}} + \sum_{\ell^w \in \omega^+(v)} x_{k\ell^D}^{\text{LOCO}} - x_{kv}^{\text{NEED}} \leq n_{k,v}^{\text{SPARE}} \qquad k \in K_r, v \in V^{\text{SRC}} \tag{2}$$

$$\sum_{s \in S_v^+} n_k^s\, z_s + \sum_{\ell^w \in \omega^+(v)} x_{kL^{\text{W}}}^{\text{LOCO}} + \sum_{\ell^w \in \omega^+(v)} x_{k\ell^D}^{\text{LOCO}} \leq n_{k,v}^{\text{SPARE}} \qquad k \in K_c, v \in V^{\text{SRC}} \tag{3}$$

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

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

$$\sum_{s \in S_v^+} n_{k_r}^s\, z_s + \sum_{\ell \in \omega^+(v) \,\cap\, (L^{\text{WAIT}} \cup L^D)} x_{k_r\ell}^{\text{LOCO}}$$
$$= \sum_{s \in S_v^-} n_{k_r}^s\, z_s + \sum_{\ell \in \omega^-(v) \,\cap\, L^{\text{SHOP}}} x_{k_c\ell}^{\text{LOCO}} + \sum_{\ell \in \omega^-(v) \,\cap\, (L^{\text{WAIT}} \cup L^D)} x_{k_r\ell}^{\text{LOCO}}$$
$$v \in \delta^+(L^{\text{SHOP}}), k = \{k_r, k_c\} \in K \tag{6}$$

$$\sum_{s \in S_v^+} n_k^s z_s + \sum_{\ell \in \omega^+(v) \,\cap\, (L^{\text{WAIT}} \cup L^D)} x_{k\ell}^{\text{LOCO}} = \sum_{s \in S_v^-} n_k^s z_s + \sum_{\ell \in \omega^-(v) \,\cap\, (L^{\text{WAIT}} \cup L^D)} x_{k\ell}^{\text{LOCO}}$$
$$v \in \delta^+(L^{\text{SHOP}}), k \in K_c. \tag{7}$$

$$\sum_{s \in S_v^+} n_k^s\, z_s \leq \sum_{\ell^w \in \omega^-(v) \,\cap\, L^{\text{WAIT}}} x_{k\ell}^{\text{LOCO}} \qquad v \in V \setminus \left(V^{\text{SRC}} \cup v^{\text{SINK}} \cup \delta^+(L^{\text{SHOP}})\right),$$
$$k \in K_r \cup K_c \tag{8}$$

$$\sum_{s \in S_v^+} n_{k_r}^s\, z_s \leq \sum_{\ell \in \omega^-(v) \,\cap\, L^{\text{SHOP}}} x_{k_c\ell}^{\text{LOCO}} + \sum_{\ell \in \omega^-(v) \,\cap\, L^{\text{WAIT}}} x_{k_r\ell}^{\text{LOCO}} \qquad v \in \delta^+(L^{\text{SHOP}}), k \in K \tag{9}$$

$$\sum_{s \in S_v^+} n_k^s z_s \leq \sum_{\ell \in \omega^-(v) \,\cap\, L^{\text{WAIT}}} x_{k\ell}^{\text{LOCO}} \qquad v \in \delta^+(L^{\text{SHOP}}), k \in K_c \tag{10}$$

$$\sum_{s \in S} d_\ell^s \cdot z_s = 1 \qquad \ell \in L^T \tag{11}$$

$$\sum_{k \in K} x_{k_c \ell^{\text{SHOP}}}^{\text{LOCO}} \leq \text{CAP}(\ell^{\text{SHOP}}) \qquad \ell^{\text{SHOP}} \in L^{\text{SHOP}} \tag{12}$$

$$\sum_{k \in K} x_{kL^{\text{W}}}^{\text{LOCO}} = 0 \qquad \ell^w \in L^{\text{W\_IN}} \setminus \omega^+(V^{\text{SRC}}) : \mathsf{time}(\ell^w) < \mathsf{dwell\_loco}. \tag{13}$$

Constraints (2) and (3) guarantee that we do not exceed the number of spare locomotives, or, if we do it, it is with the minium number of additional (regular) locomotives, thanks to the minimization of the third term in the objective.

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, before either renting locomotives, or delaying a train.

Constraints (5), (6) and(7) are the flow conservation constraints for normal nodes and shop end nodes, excluding the source and dummy sinking nodes. Note that critical locomotives are relabelled as regular after completing the maintenance process at a shop node. Constraints (8), (9), (10) take care of that relabelling in the flow conservation constraints. Constraints (11) guarantee that each train should belong to **exactly** one consist travel planin the locomotive assignment. Constraints (12) limit the number of critical locomotives at any given time in a maintenance shop. Constraints (13) guarantee that between any two consecutive consist travel plans, there is a dwell time of at least dwell_loco (set to 2 hours in this study), for the time required to bust and re-assemble locomotive consists.

4 Solution Process

4.1 CG Decomposition

The model described in the previous section, called Restricted Master Problem (RMP), is first solved with an initial limited number of consist travel plans. A consist travel plangenerator, so-called pricing problem in optimization, see, e.g., Chvátal *et al.* [5], create an improving column, i.e., a consist travel planwhose addition improves the value of the linear relaxation of the current restricted master problem, or concludes that the current solution of the RMP is indeed the optimal solution of the linear relaxation of RMP. It then remains to generate an integer solution, which can be done using an iterative rounding off procedure. Such a procedure has proved to be effective in order to reach accurate ILP solutions, see the numerical results in Section 5.2. So we use the simple rounding of process instead of developing a more computational costly ILP solution method.

4.2 Enhanced Pricing Problem: Multiple Column Generation Based on Reduced Network

For the CG decomposition process described in [10], the proposed algorithm can reach an optimal solution but the time and space requirements are still high for large scale data sets. For further improvement of CG, we introduce a key feature: a reduced network for each train, which is the set of trains that can be connected by the waiting links, i.e., those trains can be assigned to a consist travel planor consist travel plan. There is a two-stage pre-process to get the reduced network of each train: firstly to cut off the un-used links from the time-space network architecture for the given train, based on the time limitation, and then to remove un-connectable trains.

In the CG process used in [10], each pricing problem (PP) (and there are as many as the number of possible origins for a consist) uses the same data set of RMP and the same set of dual values. In the newly proposed LAP model, we introduce the flexibility for each PP to choose any train source node as the origin to build the consist travel plan and generate a new column for RMP. While not fixing the origin of the consist generated in a given PP offers more flexibility, it leads to more computational expensive PPs. However, this is counterbalanced by the reduced time for checking the optimality condition. Observe that in the original CG algorithm of [10], while each PP takes significantly much less time for its solution, satisfying the optimality condition for an optimal LP solution requires solving a whole sequence of PPs without being able to generate a single consist with a negative reduced cost: this this is computational costly. Observe that we can generate more than one consist with a negative cost when solving a given PP. In practice, we generated the one with the most negative reduced cost, as well as potentially several other ones. // Re-using the idea of reduced network and of conflict graphs introduced in [10], we wrote a more generic PP than in [10]. Indeed, instead of rooting each PP to a given train, the generic PP selects the leading train of the generated consist, and generates the associated reduced network. A major advantage of such a PP leads to a much faster way to check whether the LP optimality condition is satisfied,as it requires the solution of a single PP.

In order to use the last feature in the newly proposed column generation, instead of considering the overall set of trains in the multi-commodity network, we divide it (but not with a partition scheme) into several overlapping reduced networks. Indeed, we break the original network around some critical trains, and consider each time two cases, whether the consist will use or not those critical trains. If we allow the consist to use a given critical train, we eliminate those trains that are unreachable within a consist, leading to a reduced graph. In order to generate a reduced network with only denied trains, the set of critical trains is selected in such a way that, in any optimal solution, one of the critical train has to be selected and embedded in a consist. This way, we generate a set of reduced balanced networks, that are usually not train disjoint.

4.3 Pricing Problem: Multiple Consist Travel PlansGenerator

4.3.1 Multi-CG Model

Variables.
$\text{SRC}_v \in \{0,1\}$. SRC_v is equal to 1 if a consist travel planunder construction starts at node v, 0 otherwise, for $v \in \delta^-(L_c^T)$. And same situation is applied to DST_v for its end node.
$x_\ell \in \{0,1\}$. :x_ℓ is equal to 1 if link $\ell \in L^T \cup L^W$ belongs to the path supporting any of the consist travel plans, 0 otherwise.
$n_\ell^k \in Z_0^+$. It defines the number of locomotives of type k going through $\ell \in L^T \cup L^W$. Note that $n_{\ell^t}^k > 0$ if and only if $x_{\ell^t} = 1$, but there is no such limit for $n_{\ell^w}^k$.
Observe that the flow decision variables x_ℓ and n_ℓ^k have no path index, as paths (consist travel plans) are node-disjoint.

Objective.

$$\begin{aligned}
\overline{\text{COST}} = & \sum_{k \in K^r \cup K^c} \sum_{\ell \in \omega^-(v^{\text{SINK}}) \cap L^W} n_\ell^k \\
& - \sum_{k \in K_r \cup k_c} \sum_{v \in V \setminus (V^{\text{SRC}} \cup \{v^{\text{SINK}}\} \cup \delta^+(L^{\text{SHOP}}))} u_{kv}^{(5)} \cdot \left(\sum_{\ell \in \delta-(v)} n_\ell^k - \sum_{\ell \in \delta+(v)} n_\ell^k \right) \\
& - \sum_{k \in K_r} \sum_{v \in \delta^+(L^{\text{SHOP}})} u_{kv}^{(6)} \cdot \left(\sum_{\ell \in \delta-(v)} n_\ell^k - \sum_{\ell \in \delta+(v)} n_\ell^k \right) \\
& - \sum_{k \in K_c} \sum_{v \in \delta^+(L^{\text{SHOP}})} u_{kv}^{(7)} \cdot \left(\sum_{\ell \in \delta-(v)} n_\ell^k - \sum_{\ell \in \delta+(v)} n_\ell^k \right) \\
& + \sum_{k \in K_r \cup k_c} \sum_{v \in V \setminus (V^{\text{SRC}} \cup \{v^{\text{SINK}}\} \cup \delta^+(L^{\text{SHOP}}))} u_{kv}^{(8)} \cdot \sum_{\ell \in \delta+(v)} n_\ell^k \\
& \quad + \sum_{v \in \delta^+(L^{\text{SHOP}})} u_{k,v}^{(9)} \cdot \sum_{\ell \in \delta+(v)} n_\ell^k + \sum_{v \in \delta^+(L^{\text{SHOP}})} u_{k,v}^{(10)} \cdot \sum_{\ell \in \delta+(v)} n_\ell^k - \sum_{\ell \in L^T} u_\ell^{(11)} \cdot x_\ell. \qquad (14)
\end{aligned}$$

Constraints.

$$\sum_{v \in \delta^-(L^T \cap c)} \text{SRC}_v \leq 1 \qquad c \in C \tag{15}$$

$$\sum_{v \in \delta^+(L^T)} \text{DST}_v = \sum_{v \in \delta^-(L^T)} \text{SRC}_v \tag{16}$$

$$\sum_{\ell \in \omega^+(v)} x_\ell - \sum_{\ell \in \omega^-(v)} x_\ell = -\text{DST}_v \qquad v \in \delta^+(L^T) \tag{17}$$

$$\sum_{\ell \in \omega^+(v)} x_\ell - \sum_{\ell \in \omega^-(v)} x_\ell = \text{SRC}_v \qquad v \in \delta^-(L^T) \tag{18}$$

$$\sum_{\ell \in \omega^+(v)} x_\ell - \sum_{\ell \in \omega^-(v)} x_\ell = 0 \qquad v \in V \setminus \left(\delta^+(L^T) \cup \delta^-(L^T) \cup V^{\text{SRC}} \cup v^{\text{SINK}}\right) \tag{19}$$

$$\sum_{\ell \in \omega^-(v) \,\&\, \ell \notin L^T} x_\ell = 0 \qquad v \in V^{\text{SRC}} \tag{20}$$

$$\sum_{\ell \in \omega^+(v^{sink}) \,\&\, \ell \notin L^T} x_\ell = 0 \tag{21}$$

$$x_\ell \geq \text{SRC}_{\delta+(\ell)} \quad ; \quad x_\ell \geq \text{DST}_{\delta-(\ell)} \qquad \ell \in L^T \tag{22}$$

$$\sum_{k \in K_r} \text{HP}_k \cdot n_\ell^k + \sum_{k \in K_c} \text{HP}_k \cdot n_\ell^k \geq x_\ell \cdot \text{HP}_t \qquad \ell (\equiv t) \in L^T \tag{23}$$

$$\sum_{k \in K_r \cup K_c} n_\ell^k \leq M \cdot x_\ell \qquad \ell \in L \setminus \left(\omega^+(V^{\text{SRC}}) \cup \omega^-(v^{\text{SINK}})\right) \tag{24}$$

$$\sum_{k \in K_r \cup K_c} \sum_{\ell \in \omega^-(v) \cap \omega^+(v') \cap L^W} n_\ell^k \leq M \cdot \text{SRC}_v \qquad v \in V \setminus V^{\text{SRC}}, v' \in V^{\text{SRC}} \tag{25}$$

$$\sum_{k \in K_r \cup K_c} \sum_{\ell \in \omega^-(v^{\text{SINK}}) \cap \omega^+(v) \cap L^W} n_\ell^k \leq M \cdot \text{DST}_v \qquad v \in V \setminus v^{\text{SINK}} \tag{26}$$

$$\sum_{\ell \in \omega^-(v) \cup L^W \cup L^T} n_\ell^k = \sum_{\ell \in \omega^+(v) \cup L^W \cup L^T} n_\ell^k \qquad v \in V \setminus \left(V^{\text{SRC}} \cup v^{\text{SINK}}\right), k \in K_r \cup K_c \tag{27}$$

$$\text{consist\_size}_{\min} \leq \sum_{k \in K_r} n_\ell^k + \sum_{k \in K_c} n_\ell^k \leq \text{consist\_size}_{\max} \qquad \ell \in L^T. \tag{28}$$

Constraints (15) allow no more than 1 source node per conflict graph. Constraints (16) guarantee that source nodes and destinations are the same amount. By these two sets, we allow that in the pricing problem, each conflict graph has at most one complete consist travel planwhich the source and destination nodes are both in it. So there is no complete consist travel planin the intersection of any two or more graphs. But multiple destination nodes are accepted in a graph. Constraints (17) are the flow conservation constraints for dvar x_ℓ, work only on train source nodes, considering the source node dvar SRC_v. Constraints (18) are the flow conservation constraints for dvar x_ℓ, work on train destination nodes, considering the source node dvar DST_v. Constraints (19) are the flow conservation constraints for dvar x_ℓ, work on other nodes, except source nodes and dummy sink node, considering the source node dvar DST_v. Constraints (20) & (21) guarantee that the path/consist travel plancan neither start from a station source node, nor end at the dummy sink node, otherwise the model can build a path without letting any $\text{SRC}_v = 1$ or $\text{DST}_v = 1$, based on the fact that station source nodes and dummy sink node are artificial nodes and no train can use them as sourcedestination nodes. Constraints (22) guarantee the train which source/destination node is the start/end of a consist travel planmust be selected. Note that in our time-space networks, trains are node-disjoint. The first flow conservation set above, will generate some paths, at most one per graph, and totally node-disjoint. This is the base for the next step to assign locomotive flows over them without the path indices. Constraints (23) assign enough power to each train selected by any consist travel plan. Constraints (24) guarantee that the power should be only assigned to the paths we selected by the first set of flow conservation constraints. Note that this set does not take effect on the waiting links from or to the artificial nodes (station source node or dummy sink node). Constraints (25) allows only the locomotive flows on the waiting links from the station source nodes to the source nodes of selected path(s). Constraints (26) allows only the locomotive flows on the waiting links from the destination nodes of selected paths to the dummy sink node. Constraints (27) are the flow conservation constraints.Constraints (28) set the upper and lower bounds of consist size. The second set of flow conservation constraints build the locomotive flows in order to assign proper locomotives to each train selected by the first flow conservation constraint set. Since the paths are node-disjoint, the flows do not need path indices. In addition, the flows are only half-limited over paths: the paths must be covered, but it is possible to use unselected waiting links to finish a flows from the "station" source node to the dummy sink node.

5 Numerical Results

The primary objective of this study is to provide a new optimization model and algorithm for the real-life locomotive assignment problem. For this reason, computational results are restricted to the CPR data sets, except for the larger data sets, which we generated to add connectable and feasible trains to the existing train schedule. We now describe the data used in the computational experiments, followed by a summary of computational results, and a comparison with our previous LAP algorithm [10].

5.1 Data Instances

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 requested by the mathematical model, the number of types was doubled in order to distinguish the critical (about 20% of the overall number of locomotives) from the non critical locomotives.

Data sets (adapted from CPR data sets) contain 862-train schedules over a time period of 7 days and 1,750-train schedule over a time period of 14 days. The maximum time period is set to two weeks, as it offers for flexibility a better planning, taking into account the overall travel times from a side of the railway network (e.g., Vancouver) to the other side (e.g., New-York). Indeed, a regular coast-to-coast train takes 5-7 days from East to West, so a two week schedule allows the planning of a round trip for a set of train pair. For larger data sets, we added 100 and 200 trains to each original CPR data set, while making sure that they can share some consists with some of the other trains.

We run numerical experiments on an adapted train scheduling data and railway infrastructure of CPR. Data include train departure times and stations, train arrival times and stations, and horse-power requirements. The railway infrastructure of CPR includes CPR's entire railway network (from Vancouver to Montreal, covering all of Canada and parts of the United States), the type of locomotives in operation, and the location and capacity of maintenance shops.

Programs/algorithms were run using CPlex 12.6.1 on a server with 40-cores, 1TB memory.

Table 1 Computational Comparison of the Different CG Model/Algorithm

# of Trains	LAP Model	LP Obj.	Total Time	ILP Obj.	# Columns Generated	# Columns Selected	Round	Req. Loc.	GAP (%)
7-day, 862 trains	SCG	630,182	20h19m36s	637,980	2,207	516	4	962	1.24
	SCG+		11h03m21s						
	MCG	630,898	1h36m02s	635,500	1,543	506	155	961	0.73
+ 100 trains	SCG	675,010	23h22m53s	683,280	2,387	534	4	1,014	1.23
	SCG+		13h32m22s						
	MCG	674,918	2h13m05s	681,360	2,027	521	203	1,006	0.95
+ 200 trains	SCG	729,404	26h34m12s	737,320	2,507	541	4	1,129	1.09
	SCG+		14h38m55s						
	MCG	730,585	3h08m55s	735,520	2,325	533	233	1,109	0.68
14-day 1,750 trains	SCG	1,057,813	44h41m53s	1,077,100	1,835	1,294	4	1,289	1.82
	SCG+		26h29m31s						
	MCG	1,057,949	11h18m55s	1,071,300	1,263	1,284	127	1,290	1.26
+ 100 trains	SCG	1,113,992	50h58m21s	1,126,780	1,654	1,355	4	1,350	1.15
	SCG+		28h19m45s						
	MCG	1,113,926	12h50m21s	1,129,860	1,544	1,355	155	1,350	1.43
+ 200 trains	SCG	1,165,227	56h19m41s	1,178,880	1,853	1,345	4	1,464	1.17
	SCG+		31h17m56s						
	MCG	1,164,679	21h27m17s	1,185,140	1,727	1,317	173	1,481	1.76

5.2 Computational Comparison of the Different CG Models/Algorithms

In Table 1, we compare different solution scenarios, for each data set: the original CG algorithm of [10] (marked as LAP-SCG), the original CG algorithm with PPs using the concept of conflict graphs (marked as LAP-SCG+), and the newly proposed multiple column generated (marked as LAP-MCG, and we set 10 columns per call of PP).

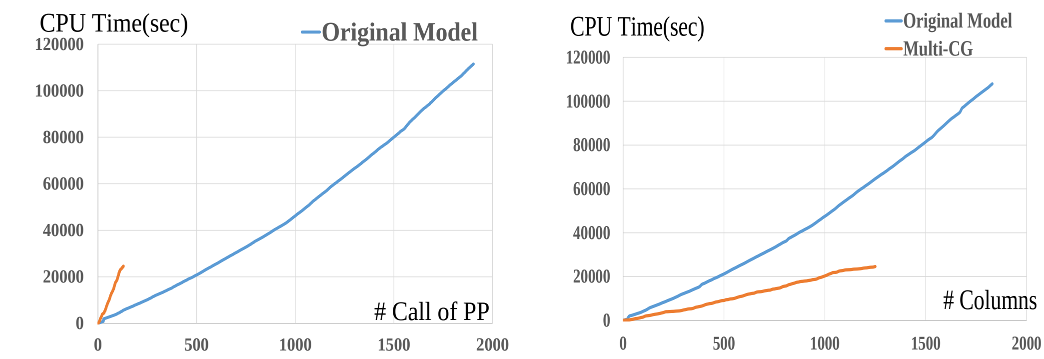
In Table 1, we provide the total computational times, the objective value of LP and ILP, the number of columns that all PPs generated throughout the overall solution process, and the number of columns selected in the final ILP solutions. The column rounds shows how many times that all possible nodes are checked as the origin of PP. The second last column contains the size of locomotive fleet that the final ILP solution recommends. The last column shows the gap between the optimal LP solution and the ε-optimal ILP solution.

We can also observe that gap is very small in practice, close to 1%, sometimes even smaller than 1%, meaning that the current multi-column algorithm outputs very good ε-optimal ILP solution.

From Table 1, we can see that using the concept of conflict graphs can reduce the total computational time by about half its value. The multi-CG algorithm can save another half of the computational times, without reducing the quality of the final ε-optimal ILP solution.

5.3 Analysis of Multi-CG Architecture

From Table 1, we can see that the Multi-CG model reduces the computational times by about 70%. The first reason is that for each PP call, the Multi-CG algorithm converges faster than the former CG from [10]: the Multi-CG model can select the consist with the best origin node rather than comparing all best consists with a fixed origin node. The second reason comes from allowing each PP call of the Multi-CG algorithm to generate several columns.



(a) CPU time vs. each PP solution (b) CPU time vs. column generation

Figure 2 Analysis of the computational computing times (CPU).

The third reason is that, even if each PP requires a longer computational time in the Multi-CG algorithm, as showed in Figure 2(a) it can generate up to 10 columns, so the time per column is much less, as showed in Figure 2(b).

6 Conclusions

The key contributions of the paper is a new CG architecture with the generation of multiple columns per pricing problem, which can greatly reduce the CPU time of the original LAP model. This multi-CG architecture can be used in CG associated with network flow formulations for networks that can be decomposed into semi-independent conflict graphs with some small overlapping. The multi-CG algorithm significantly increases the convergence rate and decreases the average generation time per column, so reduces the total time for reaching the final optimal or ε-optimal solution.

Acknowledgments. Special thanks to Mr. Peter Finnie, who kindly provides data set as the former manager of yard metrics, capital projects, costing and operations research in CPR.

References

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

2 R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.

3 V. Cacchiani, A. Caprara, and P. Toth. Models and algorithms for the train unit assignment problem. In R. Mahjoub, V. Markakis, I. Milis, and V.T. Paschos, editors, *Combinatorial Optimization*, volume 7422 of *Lecture Notes in Computer Science*, pages 24–35. Springer, 2012.

4 S. Chen and Y. Shen. An improved column generation algorithm for crew scheduling problems. *Journal of Information and Computational Science*, 10(1):175–183, 2013.

5 V. Chvátal. Linear programming. A series of books in the mathematical sciences, 1983.

6 J.-F. Cordeau, F. Soumis, and J. Desrosiers. A benders decomposition approach for the locomotive and car assignment problem. *Transportation science*, 34(2):133–149, 2000.

7 G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems*. Springer, 2002.

8 A. Fügenschuh, H. Homfeld, A. Huck, A. Martin, and Z. Yuan. Scheduling locomotives and car transfers in freight transport. *Transportation Science*, 42(4):478–491, 2008.

9 J.-L. Goffin and J.-P. Vial. Multiple cuts in the analytic center cutting plane method. *SIAM Journal on Optimization*, 11(1):266–288, 2000.

10 B. Jaumard, H. Tian, and P. Finnie. Locomotive assignment under consist busting and maintenance constraints. *Submitted*, 2014. `https://www.gerad.ca/en/papers/G-2014-54`.

11 B. Jaumard, H. Tian, and P. Finnie. Best compromise in deadheading and locomotive fleet size in locomotive assignment. In *2015 Joint Rail Conference*, pages V001T04A003–V001T04A003. American Society of Mechanical Engineers, 2015.

12 A Mingozzi, M. A. Boschetti, S. Ricciardelli, and L. Bianco. A set partitioning approach to the crew scheduling problem. *Operations Research*, 47(6):873–888, 1999.

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

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

15 M. Saddoune, G. Desaulniers, I. Elhallaoui, and F. Soumis. Integrated airline crew pairing and crew assignment by dynamic constraint aggregation. *Transportation Science*, 46(1):39–55, 2012.

16 R. Sadykov, F. Vanderbeck, A. Pessoa, and E. Uchoa. Column generation based heuristic for the generalized assignment problem. *XLVII Simpósio Brasileiro de Pesquisa Operacional, Porto de Galinhas, Brazil*, 2015.

17 C. Surapholchai, G. Reinelt, and H. G. Bock. Solving city bus scheduling problems in bangkok by eligen-algorithm. In *Modeling, Simulation and Optimization of Complex Processes*, pages 557–564. Springer, 2008.

18 B. Vaidyanathan, R.K. Ahuja, J. Liu, and L.A. Shughart. Real-life locomotive planning: new formulations and computational results. *Transportation Research*, Part B 42:147–168, 2008.

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