

# A Dynamic Row/Column Management Algorithm for Freight Train Scheduling\*

Brigitte Jaumard<sup>1</sup>, Thai H. Le<sup>1</sup>, Huaining Tian<sup>1</sup>, Ali Akgunduz<sup>2</sup>, and Peter Finnie<sup>3</sup>

- 1 Concordia University, CSE – Computer Science and Software Eng., Concordia University, Canada, [bjaumard@cse.concordia.ca](mailto:bjaumard@cse.concordia.ca)
- 2 Concordia University, MIE - Mechanical and Industrial Eng., Concordia University, Canada
- 3 CPR, Canadian Pacific Railway, Calgary, Canada

---

## Abstract

We propose a new dynamic row/column management algorithm for freight train scheduling in a single track railway system. While many papers have already been devoted to train scheduling, previously published optimization models still suffer from scalability issues, even for single track railway systems. Moreover, very few of them take into account the capacity constraints, i.e., the number of alternate tracks in the railway stations/sidings in order for the trains to meet/bypass. We propose an optimization model which takes such constraints into account, while still handling efficiently the other meaningful constraints. We design an original solution scheme with iterative additions/removals of constraints/variables in order to remain with a manageable sized mixed integer linear program at each iteration, without threatening to reach the optimal solution.

Numerical results are presented on several data instances of CPR (Canadian Pacific Railway) on the Vancouver–Calgary corridor, one of the most busy corridor in their railway system. Therein, the proposed model and algorithm are used as a planning tool to evaluate the network capacity, i.e., how much the number of trains can be increased without impacting significantly the average travel times between the source and destination stations of the various trains in the corridor. Larger data instances than those previously published are solved accurately ( $\epsilon$ -optimal solutions) for the schedule of freight trains.

**1998 ACM Subject Classification** G.1.6 Integer Programming, G.2.3 Applications

**Keywords and phrases** Railway optimization, Train scheduling, Single track

**Digital Object Identifier** 10.4230/OASICS.ATMOS.2012.108

## 1 Introduction

Train scheduling has already received a lot of attention, whether for passenger or freight trains. While passenger train schedules are relatively static and cyclic, and can be planned months ahead, freight train schedules are designed with a much shorter planning time period, sometimes even one day or few hours before train departures. Moreover, passenger train schedules must obey some strict time window constraints as trains must arrive and depart from stations in order for passengers to get off/on the trains according to the posted schedule. On the opposite, the schedule of the freight trains may vary according to the train lengths or loads, i.e., freight trains have a much greater variability in their speed. Lastly, the track configuration of the freight trains does not have a dedicated direction as it is often the case

---

\* This work was partially supported by CPR, NSERC and FQRNT.



for passenger trains. For all those reasons, the scheduling of freight trains is more complex than for passenger trains. While the volume of goods transport has increased over the years, extensions of railway systems are very rare because they represent major investments for railway companies or governments. Accordingly, the railways are often operating freight trains in a system that is close to saturation. It follows that a very effective planning and optimization of the rail network is needed. From now on, we focus on freight train scheduling, and refer the interested reader to, e.g., the following surveys for passenger train scheduling [1] [3].

Several optimization models, exact and heuristic algorithms have been proposed for freight train scheduling. We will now review the most recent ones. Kraay and Harker [7] proposed a Mixed Integer Linear Program (MILP) with only a subset of the constraints (dwell times, train meets and overtakes, time windows on the departure/arrival times) which does not include any capacity constraint, i.e., limit on the number of available tracks at a given station/siding. Moreover, they use heuristics in order to solve their model as their solution process is not able to scale with the large number of constraints and variables. Experiments are very limited (less than 11 stations/sidings along a single line track). A very similar MILP model was developed by Higgins, Kozan and Ferreira [5] and tested against a Tabu Search heuristic on data instances with up to 30 trains and 12 sidings. As for [7], the MILP model could not scale properly. Consequently, the authors only solve the linear relaxation of their MILP model, and use the lower bound it provides in order to evaluate the quality of their heuristic solutions. Depending on the papers, the objective varies from minimizing the tardiness of the trains or the fuel consumption for the most commonly considered ones.

A similar MILP model has been reused in [4, 9, 8, 2]. In [9], Zhou and Zhong design a branch-and-bound based heuristic and a Lagrangian relaxation lower bound in order to solve data instances with up to 30 trains and 18 stations. In [8, 2], the authors propose a vertical decomposition algorithm in order to overcome the scalability issues, i.e., dispatching the trains one by one, or one train cluster at a time in the MILP model. However, the size of successive MILP models to be solved is constantly increasing, and therefore the size of solved data instances is not much larger than those of previous studies. Note also that, in the first algorithm (*FixedPath*) of [8] and in [2], the definition of the route of a train includes whether to travel or not to travel a siding, and routes are defined at the outset (i.e., no optimization is made on which trains should travel the sidings). Track capacity constraints are enforced with flow conservation constraints which require the introduction of additional variables. In the second algorithm (*FlexiblePath*) of [8], routes are no more defined at the outset, however, several restrictions apply, in particular, two trains travelling in the same direction cannot be running at the same time on an identical segment, one train behind the other one (under some headway constraints). The authors solved data instances with 4 trains using their exact models, and then larger data instances with the help of heuristics and a parallel algorithm, i.e., data instances with up to 10 sidings and 24 trains.

The paper is organized as follows. In Section 2, we present the problem statement of the train scheduling in a single track freight train railway system, as that of CPR - Canadian Pacific Railway. The newly proposed optimization model is detailed in Section 3, with the inclusion of the capacity constraints. Solution of the optimization model, an original dynamic row and column generation/removal exact algorithm is described in Section 4. Numerical results are presented in Section 5 on several data set instances in order to evaluate the performance of the optimization model, as well as an estimation of the network capacity of a railway system, i.e., how many trains can run simultaneously in the system without unduly increasing the average travel times. Conclusions are drawn in the last section.

## 2 Problem Statement

This study considers a rail system with a single two-way track between stations or sidings, associated with a mesh network. Each track is divided into segments which are separated by sidings or stations. Tracks can be used by trains running in both directions, and trains can meet and pass at stations or sidings. Sidings are typically added to a railway line in order to allow two trains to pass one another and are the most common method used to expand capacity. Sidings are typically built long enough to permit trains to come to a full stop inside the siding while remaining clear of the switches at either end.

The proposed optimization model, which will be detailed in the next section, builds a freight train scheduling with all meaningful constraints. The input is the topology of the network as described by its set of segments and the list of trains that need to be scheduled, with their characteristics: origin/destination stations, expected departure/arrival times, average and maximum train speed. Moreover, each train has a specific priority which depends on the train series, i.e., the types of goods. It may also depend on the customer contract agreements and the train loads. We assume that the given railway network is a single track mesh network. Two trains in opposite directions are not allowed to be on the same track segment and they can meet each other only at a siding or a station. Two trains in the same direction can be running on a segment at the same time but they must maintain a safety distance, and they can pass each other only at a siding or a station.

The output of the model is a schedule for each train that specifies the departure and arrival times at each siding/station, and consequently the earliness/tardiness on the expected departure time with respect to the objective which has been set. In this study, we focus on the objective of minimizing the average travel times between departure/destination stations, while restricting the standard deviation to be below some threshold throughout a limit on the train end-to-end travel times.

## 3 Optimization Model

### 3.1 Input Parameters

Railway network parameters

$P = P^{\text{STATIONS}} \cup P^{\text{SIDINGS}}$ , indexed by  $p$ , where  $P^{\text{STATIONS}}$  is the set of station locations and  $P^{\text{SIDINGS}}$  is the set of siding locations.

$S$  Set of segments in the railway network, indexed by  $s$ . A segment is a single track between two successive locations of elements (either a station or a siding) of  $P$ . A segment  $s = [p, p']$  corresponds to an ordered pair of locations, with  $p$  traversed before  $p'$ .

Train parameters

$T$  Set of trains, indexed by  $t$

$T_p$  Set of trains which go through location  $p$

$T_s$  Set of trains which go through segment  $s$

$T_s^{\rightarrow}$  Set of  $(t, t')$  pairs of trains that travel segment  $s = [p, p'] \in S$  in the same direction

$T_s^{\leftarrow}$  Set of  $(t, t')$  pairs of trains that travel segment  $s = [p, p'] \in S$  in the opposite directions

$\text{SRC}(t)$  Departure station of train  $t$

$\text{DST}(t)$  Destination/arrival station of train  $t$

$S^t$  List of segments defining the route of train  $t$  from  $\text{SRC}(t)$  to  $\text{DST}(t)$

$\bar{d}_{\text{SRC}(t)}^t$	Expected (planned) departure time of train $t$ at its origin location
$\bar{a}_{\text{DST}(t)}^t$	Expected (planned) arrival time of train $t$ at its destination location
$\pi^t$	Priority (e.g., series number) of train $t$ in the network
$\text{PENAL}_{\text{OFFSET}_d}^t$	Penalty if train $t$ leaves the origin station before/after the planned departure time

#### Location and train parameters

$\text{dw}_p^t$	Minimum dwell time of train $t$ at location point $p$ . If $p$ is only a location that train $t$ is passing through, then $\text{dw}_p^t = 0$ .
$v_s^t$	Average speed of train $t$ on segment $s$ . It depends on many parameters, e.g., the number of locomotives, the length of the train, the series number of the train, the load of the cars, the slope of the track, etc.
$\text{CAP}_p$	The capacity, in terms of the number of tracks, of the siding located at $p$ , i.e., the number of parallel tracks, excluding the main track. For the time being, we do not take into account the length of the sidings vs. the length of the trains, and therefore assume that each track in a siding can host any train, one at a time.

We assume that all times are expressed in minutes. In order to simplify the expression of the constraints, we assume that all constraints are expressed in terms of times, meaning that the average/maximum speeds are translated into times it takes for a train to travel a given distance (e.g., segment):

$r_s^t$	Average time for train $t$ to travel segment $s = [p, p']$ with $p, p' \in P$ , i.e., $r_s^t = \text{Distance}(p, p') / (\text{Average speed of } t \text{ on } s = [p, p'])$ .
$\underline{r}_s^t$	Minimum time for train $t$ to travel segment $s = [p, p']$ with $p, p' \in P$ , i.e., $\underline{r}_s^t = \text{Distance}(p, p') / (\text{Speed Limit of } t \text{ on } s = [p, p'])$ .
$\tau_s^t$	Time required for train $t$ to travel the safety distance on segment $s = [p, p']$ .

### 3.2 Variables

The first set of variables are related to the arrival and departure times of the trains.

$d_p^t$	Departure time of train $t$ from location $p$
$\text{EARLY}_d^t$	Earliness of train $t$ at departure station
$\text{LATE}_d^t$	Lateness of train $t$ at departure station
$a_p^t$	Arrival time of train $t$ at location $p$
$\text{OFFSET}_d^t$	$= \max \{ \text{EARLY}_d^t, \text{LATE}_d^t \}$

All the above variables are real valued variables. Both arrival and departure time values will be rounded to the closest minute in practice. We use real valued variables to model them to simplify the solution of the model.

A train schedule is characterized by its arrival/departure time at every station/siding along its way from origin to destination:

$$\text{SCHEDULE}(t) = [(a_{\text{SRC}(t)}^t, d_{\text{SRC}(t)}^t), \dots, (a_p^t, d_p^t), \dots, (a_{\text{DST}(t)}^t, d_{\text{DST}(t)}^t)].$$

The next set of variables corresponds to decision variables, which takes their values in  $\{0, 1\}$ .

For any  $t, t' \in T_p^{\rightarrow} : t < t'; s = [p, p'] \in S; p, p' \in P$ :

$$\theta_p^{tt'} = 1 \text{ if } t \text{ leaves station/siding } p \text{ before } t', 0 \text{ otherwise.}$$

For any  $t, t' \in T_s^{\leftarrow} : t < t'; s = [p, p'] \in S; p, p' \in P$ :

$$\delta_s^{tt'} = 1 \text{ if } t \text{ leaves } p \text{ before } t' \text{ towards } p', 0 \text{ otherwise.}$$

For any  $t, t' \in T_p : t < t'; p \in P$ :

$$\alpha_p^{tt'} = 1 \text{ if train } t \text{ arrives after train } t' \text{ at point } p, 0 \text{ otherwise.}$$

$$\beta_p^{tt'} = 1 \text{ if train } t' \text{ departs after the arrival of train } t \text{ at point } p, 0 \text{ if train } t \text{ departs after the arrival of train } t' \text{ at point } p.$$

For any  $t, t' \in T_p; p \in P$ :

$$\gamma_p^{tt'} = 1 \text{ if the arrival time of } t \text{ in } p \text{ is between the arrival and the departure times of } t', \text{ i.e., if } a_p^{t'} \leq a_p^t \leq d_p^{t'}.$$

Indeed,

$$\gamma_p^{tt'} = \alpha_p^{tt'} \beta_p^{tt'} \quad ; \quad \gamma_p^{t't} = (1 - \alpha_p^{tt'}) (1 - \beta_p^{tt'}) = 1 - (\alpha_p^{tt'} + \beta_p^{tt'}) + \gamma_p^{tt'}. \quad (1)$$

Using (1), we can eliminate half of the  $\gamma_p^{tt'}$  variables, defining them only for, e.g.,  $t < t'$ .

### 3.3 Minimize the Train Travel Times

We look at the objective of minimizing the train travel times in order to estimate the network capacity, i.e., the maximum number of trains which can be running on the tracks without deteriorating too much the average travel times between source/destination stations. Indeed, when a railway network is overloaded, waiting times for crossing or bypassing trains at sidings, are increasing. The analytical expression of the objective can be written:

$$\frac{1}{|T|} \sum_{t \in T} \left( \pi^t (a_{\text{DST}(t)}^t - d_{\text{SRC}(t)}^t) \right) = \frac{1}{|T|} \sum_{t \in T} \left( \pi^t (a_{\text{DST}(t)}^t - \bar{d}_{\text{SRC}(t)}^t) \right), \quad (2)$$

assuming departure times are fixed ( $d_{\text{SRC}(t)}^t = \bar{d}_{\text{SRC}(t)}^t$ ). A possible drawback of the above objective (2), i.e., the average of the train travel times, is not to be able to distinguish between, e.g., 5+5 and 2+8 (same average), resulting in large variance values in the second case. In order to overcome a possible large variance, we may impose some limit on the travel time of all or on some of the trains:

$$a_{\text{DST}(t)}^t - d_{\text{DST}(t)}^t \leq \max\_travel\_time^t, \quad (3)$$

where  $\max\_travel\_time^t$  may depend on the train priorities.

In order to evaluate the network load of a train system, it might be of interest to allow some offset times on the planned departure times. In such a case, we keep the same objective (left-hand side of inequality (2)), but add the following constraints (similar constraints could be added with respect to some arrival offset times):

For each train  $t \in T$ ,

$$d_{\text{SRC}(t)}^t = \bar{d}_{\text{SRC}(t)}^t + \text{LATE}_d^t - \text{EARLY}_d^t \quad ; \quad \text{EARLY}_d^t \leq \text{EARLY}_d^{t,\max} \quad ; \quad \text{LATE}_d^t \leq \text{LATE}_d^{t,\max}. \quad (4)$$

### 3.4 Constraints

In order for a train schedule to be feasible, each train must satisfy constraints, namely travel and dwell time constraints, safety distance constraints, segment conflict constraints, and capacity constraints. We next describe in detail these constraints.

#### 3.4.1 Travel and Dwell Time Constraints

A train may have to stop at a location point  $p \in P$  for a given dwell time  $\text{dw}_p^t$  (e.g., time for loading/unloading goods or train refuelling and crew shift). Inequality (5) guarantees that the difference between the departure and arrival times should not be smaller than the minimum dwell time.

$$d_p^t - a_p^t \geq \text{dw}_p^t \quad t \in T, p \in P. \quad (5)$$

In any case,  $d_p^t - a_p^t \geq 0$  for a location  $p \in P$  where  $t \in T$  passes through.

Since the speed of each train  $t$  is limited, it requires a minimum amount of time  $r_s^t$  to travel segment  $s = [p, p']$ . Inequality (6) ensures that the time travel of segment  $s$ , which is

$a_{p'}^t - d_p^t$ , must be at least the minimum  $\underline{r}_s^t$  time required for this segment (i.e., train must observe the speed limit).

$$a_{p'}^t - d_p^t \geq \underline{r}_s^t \quad t \in T, s = [p, p'] \in S^t, p \in P, p' \in P. \quad (6)$$

### 3.4.2 Safety Distance Constraints

Due to safety regulation, two trains that travel the same segment  $s = [p, p']$  (in the same direction) must maintain a *safety distance* (or *headway*) between them. We express this safety distance in time terms, using the average speed of the train. In order to ensure that safety distances are respected, we need to ensure that arrival and departure times are spaced far enough apart from each other. It leads to the following set of constraints:

For all  $s = [p, p'] \in S^t \cap S^{t'}; t, t' \in T_s^{\rightarrow} : t < t'; p, p' \in P$

$$d_p^t - d_p^{t'} \geq \tau_p^{t'} - M(1 - \theta_p^{tt'}) \quad a_{p'}^{t'} - a_{p'}^t \geq \tau_p^{t'} - M(1 - \theta_p^{tt'}) \quad (7)$$

$$d_p^t - d_p^{t'} \geq \tau_p^t - M\theta_p^{tt'} \quad a_{p'}^t - a_{p'}^{t'} \geq \tau_p^t - M\theta_p^{tt'}. \quad (8)$$

Indeed, if train  $t$  departs from  $p$  before  $t'$  (i.e.,  $\theta_p^{tt'} = 1$ ), then we must have  $d_p^t - d_p^{t'} \geq \tau_p^{t'}$ , as well as  $a_{p'}^{t'} - a_{p'}^t \geq \tau_p^{t'}$ , see (7) when the speed of  $t'$  is higher than the speed of  $t$  over segment  $s$ , as otherwise the constraint is always satisfied as long as departure times are sufficiently spaced out. In such a case, constraints (8) are redundant. On the other hand, if train  $t'$  departs from  $p$  before  $t$  (i.e.,  $\theta_p^{tt'} = 0$ ), the meaningful constraints are (8), while constraints (7) are redundant.

### 3.4.3 Segment Conflict Constraints

For two trains  $t$  and  $t'$  which need to go through a given single track segment in opposite directions, we must ensure that one train at most is running on the segment. This is the purpose of the next set of constraints:

For all  $s = [p, p'] \in S; t, t' \in T_s^{\leftrightarrow} : t < t'; p, p' \in P$

$$a_p^{t'} \leq d_p^t + M(1 - \delta_s^{tt'}) \quad ; \quad a_p^t \leq d_p^{t'} + M\delta_s^{tt'}. \quad (9)$$

Indeed, for a given segment  $s = [p, p']$ , with  $s \in S^t$  and  $s' = -s = [p', p] \in S^{t'}$ , and two trains  $t, t'$  such that  $t < t'$ , then either train  $t$  reaches  $p'$  before train  $t'$  departs from  $p'$  (i.e.,  $\delta_s^{tt'} = 1$ ), or train  $t'$  reaches  $p$  before train  $t$  departs from  $p$  (i.e.,  $\delta_s^{tt'} = 0$ ). In the former case, we must ensure that  $a_p^{t'} \leq d_p^t$ , while in the latter case, we must ensure that  $a_p^t \leq d_p^{t'}$ .

### 3.4.4 Capacity constraints

Note that in order to reduce the number of variables and constraints, we define  $\alpha_p^{tt'}$  and  $\beta_p^{tt'}$  for  $t < t'$ , but  $\gamma_p^{tt'}$ , for all pairs of  $t, t'$ . Capacity constraints are expressed as follows. For all  $p \in P; t, t' \in T_p : t < t'$ , we have:

$$-M\alpha_p^{tt'} \leq a_p^{t'} - a_p^t \leq M(1 - \alpha_p^{tt'}) \quad (10)$$

$$-M(1 - \alpha_p^{tt'} + \beta_p^{tt'}) \leq a_p^t - d_p^{t'} \leq M(1 - \beta_p^{tt'}) \quad (11)$$

$$-M(1 - \beta_p^{tt'} + \alpha_p^{tt'}) \leq a_p^{t'} - d_p^t \leq M\beta_p^{tt'} \quad (12)$$

$$(\alpha_p^{tt'} - 1) + \beta_p^{tt'} \leq \gamma_p^{tt'} \leq \min\{\alpha_p^{tt'}; \beta_p^{tt'}\} \quad (13)$$

$$-\left((\alpha_p^{tt'} - 1) + \beta_p^{tt'}\right) \leq \gamma_p^{t't} \leq \min\{1 - \alpha_p^{tt'}; 1 - \beta_p^{tt'}\}. \quad (14)$$

In addition, we have:

$$\sum_{t' \in T_p} \gamma_p^{tt'} = \sum_{t' \in T_p: t < t'} \gamma_p^{tt'} + \sum_{t' \in T_p: t > t'} \gamma_p^{tt'} \leq \text{CAP}_p \quad p \in P; t \in T_p. \quad (15)$$

Each station/siding point  $p$  has a limited capacity, i.e., number of side tracks  $\text{CAP}_p$ , for trains to meet, overtake or platform. We need to make sure that at any given time, no more than  $\text{CAP}_p + 1$  (with one train on the main track) trains, regardless of their directions, can be at  $p$ . Recall that a train  $t$  stays at  $p$  during the interval  $[a_p^t, d_p^t]$ . Constraints (10) determine the value of variables  $\alpha_p^{tt'}$ , i.e., the order of train arrival, between  $t$  and  $t'$ , at station/siding  $p$ . Constraints (11)-(12) determine the value of variables  $\beta_p^{tt'}$ . Constraints (13) - (14) determine the values of variables  $\gamma^{tt'}$  and  $\gamma^{t't}$ , and correspond to linearization of the following quadratic terms defined in (1). Finally, inequality (15) ensures that at anytime, at most  $\text{CAP}_p + 1$  trains in any direction can be at the same crossing point  $p$ .

#### 4 Solving the STTS\_M (Single Track Train Scheduling) Model

In order to overcome the large number of constraints and variables, we propose a row and column generation algorithm, called STTS\_A and depicted in Figure 1, in which we iteratively add/remove some rows and columns until we reach an  $\varepsilon$ -optimal train schedule. Indeed, the idea is to start with a rather small optimization model made of constraints (4) - (6) only, i.e., of the constraints involving only continuous variables: the earliness and tardiness constraints (4), the dwell constraints (5), and the travel time constraints (6). Note that this first group of constraints only involves continuous variables, as it does not involve any of the decision (integer) variables  $\delta_s^{tt'}, \alpha_p^{tt'}, \beta_p^{tt'}, \gamma_p^{tt'}, \theta_p^{tt'}$ , and therefore is an easy problem to solve as it is a linear program (LP).

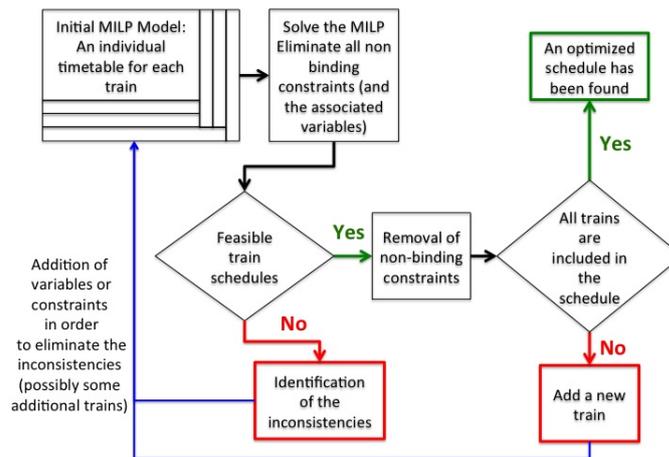
The resulting LP model is then solved, and then we check the feasibility of the solution, examining the constraints involving the interaction between two (or more) train schedules. Note that those last constraints, namely, constraints (7) up to (15), each involves one or two binary variables (with some constraints sharing the same binary variable(s)), so that their addition to the incumbent mathematical program will often entail the addition of one or two new 0-1 variables. A compromise has to be found for the number of added constraints and variables at each iteration between the following two extreme strategies: adding one violated constraint at a time or adding all violated constraints. With the first strategy, the convergence might be too slow, while with the second strategy, we might end up very quickly with an unnecessary large set of constraints and variables. Once we have added some or all violated constraints, the optimization model becomes a MILP model, which is solved again, and we keep adding violated constraints until all constraints are satisfied. Note that, in practice, it does not require solving the MILP STTS\_M model with the explicit embedding of all possible constraints, but with a quite small fraction of the overall set of variables and constraints, as will be seen in Section 5.

For the addition of the violated constraints, we consider the following strategy. Trains are ordered according to a given criterion. In this study we order the trains according to the departure times, alternating between westbound trains and eastbound trains (as the rail network we consider is an East  $\leftrightarrow$  West one). Remaining ties, if any, are arbitrarily broken. At iteration  $\text{iter} \geq 2$ , after solving the current MILP, we revisit the constraints for all the train interaction constraints, namely, (7) up to (15) with respect to the first  $\text{iter}$  trains, identify the ones which are violated and add them to the current MILP. Once we

reach iteration  $\text{iter} = |T|$ , we may need several iterations before reaching a feasible schedule, i.e., train schedules which satisfy all constraints.

After conducting some experiments, we found out that, rather than adding all violated constraints at each iteration, it was more efficient (with respect to the overall computing times) to only add the first 100 violated constraints.

Note that in the course of the iterations, we may have too many constraints and variables, so that the scalability of the current MILP is impaired. In such a case, except for constraints (4) - (6), we remove all the other constraints which are not binding constraints in the last computed MILP solution.



■ **Figure 1** Flowchart of the Solution Process.

## 5 Numerical Results

### 5.1 Data Instances

We evaluated the performance of the STTS\_M model and STTS\_A algorithm proposed in the previous sections on the Canadian Pacific Railway (CPR) network between Calgary and Vancouver [6]. It is a single track railway system, which we divided into 5 subdivisions. The number of sidings/stations in each subdivision (including the endpoints) is:

- Subdivision 1: Calgary - Field - 16 stations or sidings
- Subdivision 2: Field - Revelstoke - 13 stations or sidings
- Subdivision 3: Revelstoke - Kamloops - 14 stations or sidings
- Subdivision 4: Kamloops - Mission - 14 stations or sidings
- Subdivision 5: Mission - Vancouver - 16 stations or sidings

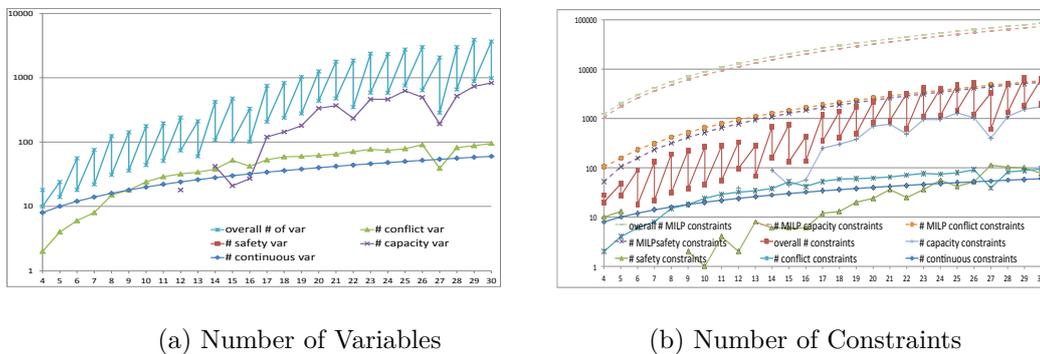
In terms of capacity (number of alternate tracks), we assume 2 alternate tracks at every location which is the endpoint of a subdivision, and 1 otherwise. The algorithm STTS\_A was run on 1 to 5 subdivisions with a variable number of trains in order to evaluate its performance, but also the network capacity of the railway system. Indeed, there is a compromise between the number of trains in the railway system and the overall travel times of the trains: if the number of trains is too large, then the overall travel times of the trains increase with significant waiting times, which is undesirable.

We use a set of a 16 to 30 trains, with 61 sidings/stations, (with the same number of trains from Vancouver towards Calgary as from Calgary towards Vancouver) with departure times uniformly distributed over a time period of 24 hours. Consequently, when the number of trains increases, their departure times are less spaced.

## 5.2 Efficiency of the STTS\_A Algorithm

Following the description of the STTS\_A algorithm in the previous section, the algorithm iteratively adds trains to be taken into account in the overall train schedule, and alternates between adding violated constraints and removing non binding constraints. First set of experiments was done with the objective 2, i.e., non flexibility on the departure times.

In Figure 2, we plot the number of constraints and variables at each major iteration (i.e., when we add a new train to be taken into account in the schedule) of the STTS\_A algorithm for train scheduling with 30 trains. We remove non binding constraints before inserting the constraints (4)-(6) related to an additional train, so we plotted the number of variables/constraints before/right after the removal of the non binding constraints for the curves associated with their overall number. Those plots correspond to the saw-tooth curves in Figure 2. In addition, we added the plots related to the number of constraints/variables for each set of constraints, but plotted only the numbers after the removal of the non binding constraints. In Figure 2, the dash lines correspond to the overall number of constraints in the MILP model: we observe that it goes beyond several tens of thousands constraints while the number of considered constraints never exceed 10,000 for 30 trains. The legend indicates the different groups of constraints.



■ **Figure 2** Evolution of the number of variables and constraints (1 subdivision).

We observe that the STTS\_A algorithm allows remaining with a highly manageable set of constraints and variables, in spite of the theoretical huge number of variables and constraints of the model, in particular when the number of trains increases. For instance, the complete MILP model contains 45,405 binary variables and 84,870 constraints for 30 trains. As expected, the dominant group of constraints corresponds to the capacity constraints as soon as the number of trains increase, while the safety constraints are much less critical (due to the distribution of train departure times).

## 5.3 Travel Times vs. Number of Trains

We now investigate the network capacity of the Calgary - Vancouver corridor, using objective (2), i.e., the average travel times. The goal is to investigate the increase of the travel times

from source to destination vs. the number of trains running in the railway network. To do so, we use the following statistics:

- Average travel times (mean -  $\mu$ , lower bound - LB, standard deviation -  $\sigma$ ):
 
$$\left( \sum_{t \in T} (a_{\text{DST}}^t - d_{\text{SRC}}^t) \right) / |T| ;$$
- Average waiting times (mean -  $\mu$ , standard deviation -  $\sigma$ ):
 
$$\left( \sum_{t \in T} \sum_{p \in P} (a_p^t - d_p^t - dw_p^t) \right) / |T| ;$$
- Number of train meetings out of the overall number of possible ones ;
- Accuracy ( $\varepsilon^{\text{OUT}}$ ) of the  $\varepsilon$ -optimal solution (relative value of the difference between the incumbent value and a lower LP bound) vs. initial requested accuracy ( $\varepsilon^{\text{IN}}$  when solving the MILP with the CPLEX solver).

■ **Table 1** Travel times vs. network load – No flexibility on departure times.

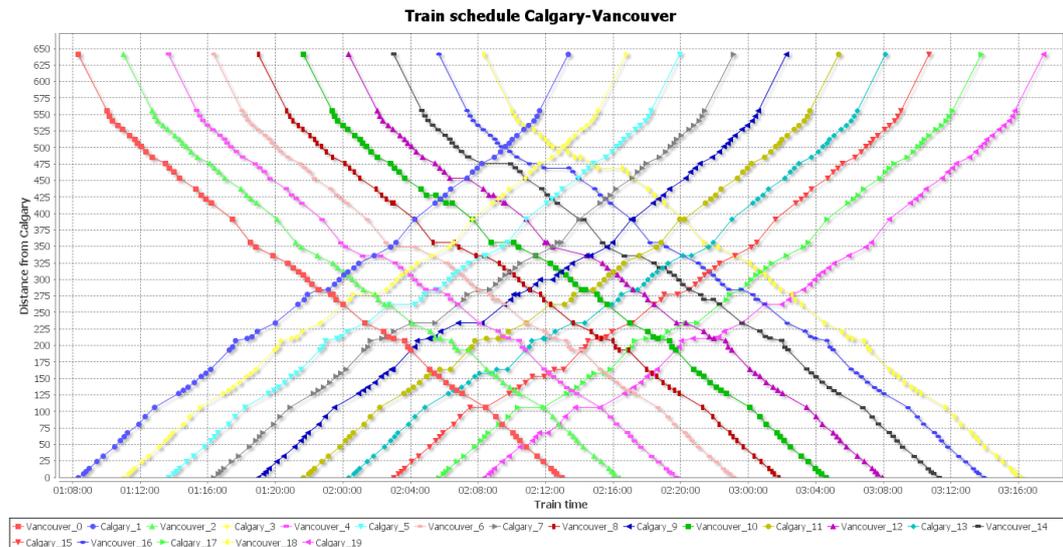
All times are in hours	T	Average travel times			Average waiting times		Number of train meetings	$\varepsilon^{\text{IN}}$	$\varepsilon^{\text{OUT}}$	travel	CPU h:m
		$\mu$	LB	$\sigma$	$\mu$	$\sigma$					
1 subdivision: Kamloops ↓ Revelstoke	16	7:08	6:59	1:03	0:45	0:53	29/64	10	2.2	7:12	00:01
	18	7:22*	7:22	0:54	1:01	0:40	39/81	10	0.0	7:24	00:01
	20	7:33	7:22	0:51	1:12	0:37	50/100	15	2.4	7:35	00:00
	22	10:46*	10:46	5:52	3:59	5:05	90/121	15	0.0	10:54	00:08
	24	11:39*	11:39	6:30	4:26	5:40	107/144	15	0.0	11:54	00:22
	28	11:40	11:32	6:26	4:35	5:46	141/196	15	1.2	11:48	01:48
30	12:16	11:54	6:22	5:29	5:54	167/225	15	3.0	12:35	04:54	
3 subdivisions: Kamloops ↓ Calgary	16	21:58	20:47	1:16	2:35	1:16	59/64	15	5.4	22:00	00:01
	18	22:02	20:59	1:38	2:18	1:29	75/81	15	4.8	22:06	00:03
	20	21:53	20:55	0:55	2:30	0:48	94/100	15	4.4	22:11	00:06
	22	22:15	21:06	2:36	1:05	2:36	115/221	15	5.1	22:18	00:06
	24	22:16	20:31	4:42	3:22	1:19	129/144	15	7.9	24:00	00:23
	26	24:19	22:08	2:00	4:07	1:47	165/169	15	8.9	24:30	00:25
28	23:51	21:47	1:46	3:31	1:36	192/196	10	8.7	26:48	11:57	
30	26:38	22:53	2:44	5:34	2:03	221/225	15	14.1	29:00	19:41	
5 subdivisions: Vancouver ↓ Calgary	16	30:08	29:27	0:54	1:22	0:53	64/64	15	2.3	30:26	00:02
	18	30:54	29:46	1:15	2:21	1:19	81/81	10	3.7	31:06	00:05
	20	31:00	29:38	1:12	2:29	1:08	100/100	15	4.4	31:06	00:03
	22	31:18	29:50	1:38	2:42	1:36	121/121	15	4.7	31:30	00:04
	24	31:45	30:03	1:32	2:31	1:12	144/144	15	5.4	31:48	00:09
	26	31:50	30:09	1:48	2:58	1:34	169/169	15	5.3	32:24	02:21
28	32:24	30:17	1:49	3:31	1:48	196/196	15	6.5	34:00	02:04	
30	34:07	31:16	2:10	4:37	1:44	225/225	15	10.4	35:47	10:57	

■ **Table 2** Travel times vs. network load – Some flexibility around the planned departure times.

# Trains	Average travel times		Number of train meetings	$\varepsilon^{\text{OUT}}$	travel	$\frac{\sum_{t \in T} \text{EARLY}_d^t}{ T }$	$\frac{\sum_{t \in T} \text{LATE}_d^t}{ T }$
	$\mu$	$\sigma$					
Trains depart on planned departure times							
16	7:08	1:03	29/64	2.2	7:12	0.	0.
18	7:22	0:54	39/81	0.0	7:24	0.	0.
20	7:33	0:51	50/100	2.4	7:35	0.	0.
Trains can be up to 30 mn early and 30 mn late with respect to planned departure times							
16	6:50	0:51	27/64	7.0	7:00	0:03 (4)	0:16 (12)
18	6:52	0:35	37/81	7.5	7:00	0:10 (7)	0:14 (11)
20	6:50	0:32	49/100	3.8	7:00	0:08 (9)	0:11 (11)

Statistics are reported for 1, 3 and 5 subdivisions, i.e., for 14, 37 and 61 sidings/stations respectively. The requested precision at the outset  $\varepsilon$  varies between 10% and 15% in order not to spend too much time solving the first MILP models. As can be observed in the column entitled  $\varepsilon^{\text{OUT}}$ , the final precision is often much better than the requested one (\* means that the optimal value has been obtained, i.e.,  $\varepsilon^{\text{OUT}} = 0$ ). However, the obtained precision varies with the number of trains and partially explains why the average times are not always strictly increasing when the number of trains is increasing for a given number of subdivisions. The optimal value is however guaranteed to lie in the interval  $\left[ \text{LB}, \sum_{t \in T} (a_{\text{DST}}^t - d_{\text{SRC}}^t) / |T| \right]$ , and there is a clear trend of increasing LB and average travel times values. The increase of the average travel and waiting times are consistent due to train meetings, as expected. Note that there is no guarantee that average travel times are always increasing when the number of trains is increasing. Consider the example with 4 stations  $p_1, p_2, p_3, p_4$  evenly spaced (40 miles for each segment), and 2 trains, one eastbound ( $t_1$ ) and one westbound ( $t_2$ ) running at 40mph. Assume  $t_1$  leaves at 8:05am from  $p_1$  and  $t_2$  leaves  $p_4$  at 8:00am. The average travel times is then 3:28 hours. Add a new westbound train leaving  $p_4$  at 7:55am with the same speed, then the average travel times becomes 3:18 hours.

In order to illustrate the train scheduling, we represented one of them with the so-called string graph for an instance with 5 subdivisions, i.e., the entire Vancouver - Calgary corridor with 20 trains. String graphs are used to display spatial and temporal information of track occupancy: the vertical axis contains the distances between the Eastern and Western stations (or the location of the intermediate sidings/stations) while the horizontal axis is a time axis.



■ **Figure 3** String graph (Vancouver – Calgary – 20 trains – 5 subdivisions).

In Table 2, we report results on how much we can improve the average travel times when allowing some flexibility on the departure times. We consider two scenarios, the first one where the trains depart on time, and a second one where trains depart with no more than 30mn early/late. Results are given for 16, 18 and 20 trains on one subdivision (Revelstoke  $\leftrightarrow$  Kamloops). We observe that a shift of a few minutes is often sufficient to reduce the number of train meets and therefore to reduce the average travel times. Numbers in parenthesis indicate the number of trains leaving earlier/later than expected.

## 6 Conclusions

Following the scarce resources of freight train companies, efficient scheduling tools are required in order to optimize the track usage, minimize the train travel times and evaluate/anticipate the saturation of a railway network. In this study, we propose an enhanced optimization model which includes the siding/station capacities, as well as an algorithm which allows a proper management of the constraints and variables in order to remain scalable even for large data instances. Indeed, it is able to solve accurately instances for up to 61 siding/stations and 30 trains within few hours.

**Acknowledgements** B. Jaumard was supported by NSERC (Natural Sciences and Engineering Research Council of Canada) and by a Concordia University Research Chair (Tier I). T.H. Le and H. Tian were each supported by a FQRNT-NSERC-CPR fellowship.

---

### References

- 1 A. Caprara, L.G. Kroon, M. Monaci, M. Peeters, and P. Toth. Passenger railway optimization. In C. Barnhart and G. Laporte, editors, *Handbooks in Operations Research and Management Science*, volume 14, chapter 3, page 129–187. Elsevier, 2007.
- 2 M. Carey and D. Lookwood. A model, algorithms and strategy for train pathing. *Journal of Operation Research Society*, 46(8):988–1005, 1995.
- 3 J.-F. Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32:380 – 404, April 1998.
- 4 M. Dessouky, Q. Lu, J. Zhao, and R.C. Leachman. An exact solution procedure for determining the optimal dispatching times for complex rail networks. *IIE Transactions*, 38:141–152, 2006.
- 5 A. Higgins, E. Kozan, and L. Ferreira. Optimal scheduling of trains on a single line track. *Transportation Research B*, 30(2):147–161, 1996.
- 6 P. Ireland, R. Case, J. Fallis, C. Van Dyke, J. Kuehn, and M. Meketon. The Canadian Pacific Railway transforms operations by using models to develop its operating plans. *Interfaces*, 34(1):5–14, 2004.
- 7 D. Kraay and P.T. Harker. Real-time scheduling of freight railroads. *Transportation Research*, 29B(3):213–229, 1995.
- 8 S. Mu and M. Dessouky. Scheduling freight trains traveling on complex networks. *Transportation Research Part B: Methodological*, 45:1103–1123, 2011.
- 9 X. Zhou and M. Zhong. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B*, 41:320–341, 2007.