

# Decomposing the Train Scheduling Problem into Integer-Optimal Polytopes

Masoud Barah

Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, TN 37996, mbarah@vols.utk.edu

Abbas Seifi

Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, Tehran, Iran, aseifi@aut.ac.ir

James Ostrowski

Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, TN 37996, jostrows@utk.edu

This paper presents conditions for which the linear relaxation for the train scheduling problem is integer-optimal. These conditions are then used to identify how to partition a general problem's feasible region into integer-optimal polytopes. Such an approach yields an extended formulation that contains far fewer binary variables. Our computational experiments show that this approach results in significant computational savings. Moreover, this approach scales well when the train scheduling problem is modeled using smaller time increments, allowing for higher fidelity models to be solved without significantly increasing the required computational time.

*Key words:* Non-periodic Train Scheduling Problem, Space-Time Network, Integer-Optimal Polytopes

---

## 1. Introduction

Train scheduling is an important problem by which one attempts to utilize the railway network efficiently since it is an expensive infrastructure with limited capacity. Non-periodic train scheduling becomes especially challenging when trying to maximize network utilization during times with high demand. In non-periodic train scheduling with fixed traverse times, all train departures from stations are scheduled by allocating a path (route) to each train considering infrastructure limitations. Some of these limitations include maintaining the headway time between two consecutive trains on the same track, dispatching rules, avoiding simultaneous track allocation to trains, and station capacities.

The non-periodic train scheduling problem has been explored for over four decades. Initially, Szpigel (1973) introduced train scheduling for a single track railway in order to minimize the total delay of all trains using a job-shop scheduling approach and proposed a branch-and-bound solution algorithm. Cai and Goh (1994) proposed a constructive heuristic algorithm for single line train scheduling by separating conflicting trains into groups. To form conflict-free groups, some trains were forced to have delays so as to achieve minimal associated delay cost. Carey

and Lockwood (1995) developed a model that determines the sequence of trains as well as their associated departure or arrival times at all stations. Alternatively, Higgins et al. (1997) formulated a single line train scheduling by initially fixing train traversal ordering and then determining their associated departure or arrival times at all stations. Brännlund et al. (1998) discretized time intervals to one minute slots and track lines into blocks. They developed a model in which each time slot in every block was assigned to, at most, one train so that the profit is maximized.

A common modeling approach to train scheduling is based on a space-time network in which the whole time horizon is discretized to equal time intervals. The main difficulty with this approach is having to deal with track capacity constraints in an efficient way. Caprara et al. (2002) formulated the arrival, departure, and track capacity constraints (known as clique constraints) into an integer programming model. They also rewrote these constraints by introducing binary variables that reflect the summation of entering arcs to each node in order to avoid selection of pairs of conflicting arcs.

Caprara et al. (2006) expanded their original mathematical model to incorporate more practical constraints in some real-world case studies. Cacchiani et al. (2010b) extended the model in Caprara et al. (2006) to include bi-directional tracks using a new constraint called crossover constraint. Alternatively, Borndörfer and Schlechte (2007) proposed another approach to deal with track capacity constraints. They used artificial arcs to connect the pairs of arcs in each track that are compatible due to headway limitations. This would prevent allocating incompatible (infeasible) arcs to train paths. Cacchiani et al. (2008) and Cacchiani et al. (2010a) developed a model in which a binary variable represents a whole train path and two paths are in conflict whenever they do not meet network capacity constraints at any line segment.

Velasquez et al. (2005) followed the approach in Brännlund et al. (1998) to generate the track capacity constraints with shorter discretized time slots in order to better represent movement of trains on the junctions. Lusby (2008) applied a similar method to that of Velasquez et al. (2005) assuming variable traverse times. Moreover, Zhou and Zhong (2007) independently proposed a mathematical formulation for generating a train schedule in which each time slot of track capacity was considered as a limited resource. A Larangian relaxation heuristic together with a branch-and-bound method were used to solve some real-world problem instances. For more detailed review on various train scheduling models and solution approaches, one may refer to Lusby et al. (2011), Cacchiani and Toth (2012), Harrod and Schlechte (2013), and Caimi et al. (2018).

In this paper, we consider a non-periodic train scheduling problem on a macroscopic level with a single track railway between stations; that is, we construct a timetable in which trains' exact arrival and departure times are determined for any station on their paths. We construct an integer programming model based upon a space-time network with discretized time intervals in each

track using the concept of multi-commodity flow (Caprara et al. (2002), Caprara et al. (2006) and Borndörfer and Schlechte (2007), Caimi et al. (2011)) and set-packing (Brännlund et al. (1998), Velasquez et al. (2005), Lusby (2008) and Caimi et al. (2011)). Complying with the literature, we define a space-time network on which we build our basic model, a combination of flow conservation and set-packing constraints together with an objective function of maximizing total profit of scheduled trains. We conduct a polyhedral study of the basic model to identify conditions for which the resulting formulation is integer-optimal. While a condition for integer optimality is not likely to be seen in practice, we can decompose a general train scheduling problem into subproblems for which this condition is met. This allows us to reformulate the basic model so that the new model has drastically fewer binary variables, a reduction by approximately a factor of track travel times. The new model significantly outperforms the basic model in our real-world instances.

The rest of this paper is organized as follows. The basic model is explained in Section 2. Section 3 is devoted to the polyhedral study of the basic model. Then, we propose a new model formulation in Section 3.1, followed by discussions on extension to the model in Section 3.2. Computational results on some real-world examples are reported in Section 4. Finally, some conclusions are discussed in Section 5.

## 2. The Basic Model

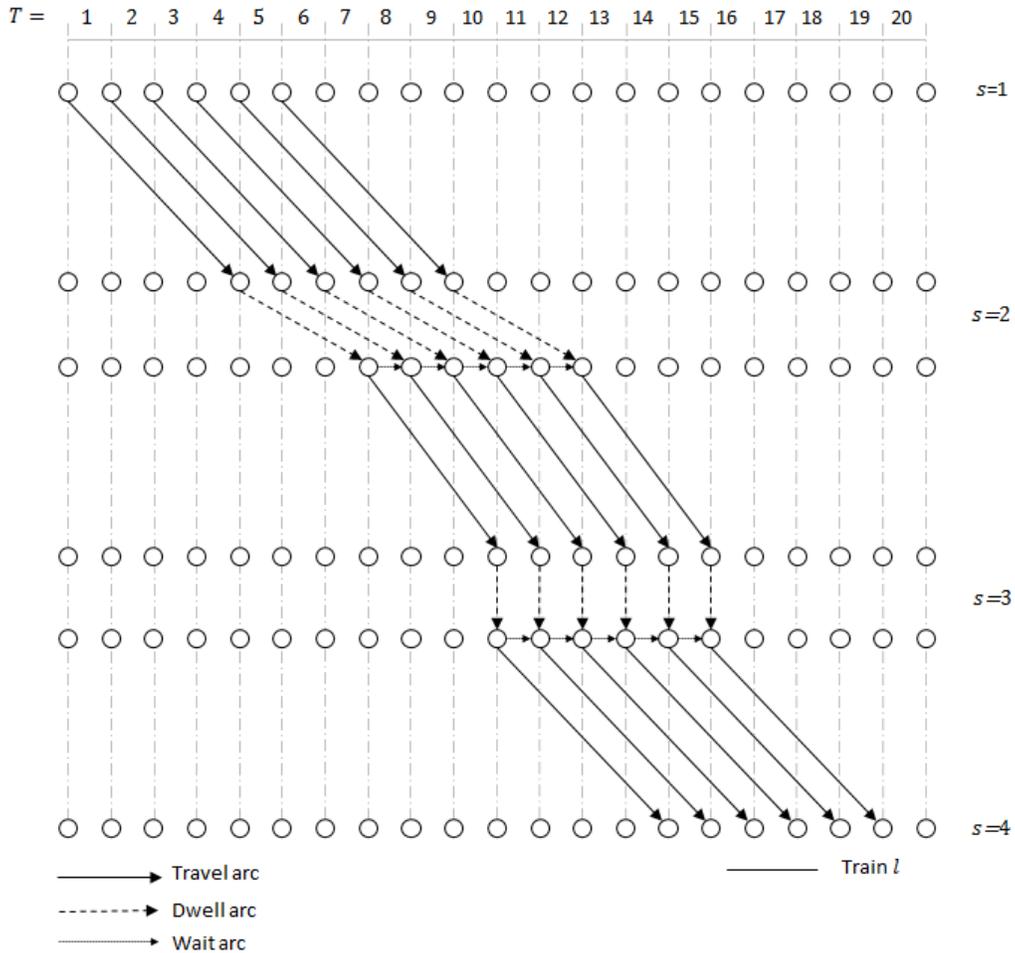
We model the infrastructure of the railway network on a macroscopic level; that is, we include stations and tracks but ignore more detailed components such as platforms and signals. Note that we make no assumptions on the topology of the network itself, however, we assume the particular path (the sequence of stations visited) across the network is known a priori for each train. We refer readers to Erol (2009) for more details on the macroscopic railway infrastructure.

Let  $L = \{1, 2, \dots, |L|\}$  be the set of trains,  $S = \{1, 2, \dots, |S|\}$  the set of stations (including artificial source station,  $\alpha$ , and artificial sink station,  $\omega$ ), and  $T = \{1, 2, \dots, |T|\}$  be the set of discrete time intervals. The list of indices and parameters are given in Tables 1 and 2 respectively.

The train scheduling problem can be visualized by in terms of flows along a digraph. We define  $D(V, A)$  as an acyclic directed network. For all stations but  $\alpha$  and  $\omega$ , we have two vertices per station per time period,  $v_{s,t}^i$  representing trains entering station  $s$  at time  $t$  and  $v_{s,t}^o$  representing trains ready to leave station  $s$  at time  $t$ . There is only one vertex representing  $\alpha$  and another single vertex representing  $\omega$  (dummy vertices).

The set  $A = A^1 \cup A^2 \cup \dots \cup A^{|L|}$  consists of the arcs obtained from the union of all possible arcs for each train in set  $L$ . As arcs are specific to each train, we can think of  $a \in A^l$  as having the color  $l$ . Each  $A^l$  is made up of three types of arcs: arcs representing travel from one station to another, arcs representing dwell time, and arcs representing waiting at a station.

Travel arcs are defined as follows. For the length of track  $(\alpha, s) \in E^l$ , we have the arc  $(\alpha, v_{s,t}^i)$  for all  $t \in [\underline{T}_\alpha, \bar{T}_\alpha]$ . Similarly, for every  $(s, \omega) \in E^l$ , we have the arc  $(v_{s,t}^o, \omega)$  for all  $t \in [\underline{T}_s, \bar{T}_s]$ . For every other length of track  $(s, s') \in E^l$  (that is, track not associated with the dummy vertices), train  $l$  must depart from station  $s$  sometime in the time interval  $[\underline{T}_s, \bar{T}_s]$ . Thus,  $A^l$  contains the arcs  $(v_{s,t}^o, v_{s',t+\epsilon_s^l}^i)$  for every  $t \in [\underline{T}_s, \bar{T}_s]$ . Dwell time occurs immediately after a train enters a station, giving the arcs  $(v_{s,t}^i, v_{s,t+D_s^l}^o)$ , also in  $A^l$ . Lastly, a train may wait at a station, represented by the arcs  $(v_{s,t}^o, v_{s,t+1}^o) \in A^l$  for all  $t \in [\underline{T}_s, \bar{T}_s - 1]$ . An example of a network  $D$  for train  $l$  is depicted in Figure 1. In this figure, the time interval associated with track  $(s = 1, s = 2)$  is  $[\underline{T}_1 = 1, \bar{T}_1 = 6]$ . The train has to dwell at station 2, whereas it has no dwell at station 3. The train gets the highest profit if it leaves station 1 at time 1 and arrives station 4 at time 15.



**Figure 1** An example of space-time network. The train visits stations  $1, \dots, 4$ , i.e.,  $S^l = \{\alpha, s_1, s_2, s_3, s_4, \omega\}$ . For ease of exposition, we have omitted artificial source,  $\alpha$ , and artificial sink,  $\omega$ , stations. The highest profit for this train is associated with the path departing station 1 at time  $t = 1$  and arriving station 4 at time  $t = 15$ .

**Table 1** The sets.

Set	Description
$T = \{1, 2, \dots,  T \}$	Set of time periods
$S = \{\alpha, 1, 2, \dots, n_S, \omega\}$	Set of stations, including artificial source $\alpha$ and sink $\omega$ , where $n_S$ denotes the number of stations in the system (excluding source and sink)
$S^l = (\alpha, s_1^l, s_2^l, \dots, s_{n_{S^l}}^l, \omega)$	Ordered sequence of stations visited by train $l$ , where $n_{S^l}$ denotes the number of stations visited by train $l$ (excluding source and sink)
$E^l = \{(s_i^l, s_{i+1}^l) \mid i \in \{1, \dots, n_{S^l} - 1\}\}$	Pairs of stations representing “from-to” actions of train $l$
$L = \{1, 2, \dots,  L \}$	Set of trains

**Table 2** The parameters.

Parameter	Description
$\alpha$	Artificial source station for all trains
$\omega$	Artificial sink station for all trains
$n_S$	Number of stations in the system (excluding source and sink)
$n_{S^l}$	Number of stations visited by train $l$ (excluding source and sink)
$D_s^l$	Required (minimum) dwell time for train $l$ at station $s$
$\epsilon_s^l$	Required (fixed) travel time for train $l$ to leave station $s$ and arrive at the next station.
$\rho_{s,t}^l$	The profit of train $l$ leaving station $s$ at time $t$
$\mu_{s,t}^l$	The penalty of train $l$ waiting at station $s$ at time $t$
$\underline{T}_s^l$	The earliest time period train $l$ can leave station $s$
$\overline{T}_s^l$	The latest time period train $l$ can leave station $s$

**Table 3** The variables.

Variable	Description
$x_{s,t}^l \in \{0, 1\}$	Binary variable representing if train $l$ leaves station $s$ at time $t$
$d_{s,t}^l \in \{0, 1\}$	Binary variable representing if train $l$ begins dwell time at station $s$ at time $t$
$w_{s,t}^l \in \{0, 1\}$	Binary variable representing if train $l$ waits at station $s$ at time $t$

There are two types of constraints, described as follows:

- Flow conservation constraints: each train should physically traverse a continuous path between its origin and destination, i.e., a path from source to sink.
- Packing constraints: no two trains can be on the same track at the same time. This include infrastructure operational constraints such as track occupancy constraints.

We solve the train scheduling problem as an integer program, where Table 3 describes the variables used in the model.

The objective function is to maximize the total profit of the scheduled trains. Let  $\rho_{s,t}^l \geq 0$  denote the profit of train  $l$  leaving station  $s$  at time  $t$ . Intuitively, train  $l$  gains the highest profit when it leaves station  $s$  at the earliest time within the departure time interval of the station, i.e.,  $t = \underline{T}_s^l$ . Accordingly, the train’s profit is non-increasing while it is delayed in leaving the station, i.e.,  $\rho_{s,t}^l \geq \rho_{s,t+1}^l$  for all  $t \in [\underline{T}_s^l, \overline{T}_s^l - 1]$ . Moreover, waiting at stations affects trains’ profit. Let  $\mu_{s,t}^l \geq 0$

denote the penalty of train  $l$  waiting at station  $s$  at time  $t$ . As  $x_{s,t}^l$  ( $w_{s,t}^l$ ) takes the value of one, if train  $l$  leaves (waits at) station  $s$  at time  $t$  and  $\rho_{s,t}^l$  ( $\mu_{s,t}^l$ ) denotes the profit (penalty) associated with train  $l$ . Then, the objective function can be written as

$$\max \sum_{l \in L} \sum_{s \in S^l} \sum_{t \in T} \rho_{s,t}^l x_{s,t}^l - \mu_{s,t}^l w_{s,t}^l. \quad (1)$$

Note that all arcs incident to source and sink stations have zero weight in the objective function. Also, note that the third sum in the objective function is over all time periods. This is for notational convenience. As the variable  $x_{s,t}^l$  can only be nonzero when  $t$  is in the interval  $[\underline{T}_s^l, \overline{T}_s^l]$ , it would be appropriate to restrict the sum just to this interval. However, for ease of exposition, we assume that variables not within this time interval are fixed to zero. That is, we have the constraints

$$x_{s,t}^l = 0 \quad \forall t \notin [\underline{T}_s^l, \overline{T}_s^l] \quad \forall l \in L, \quad s \in S^l. \quad (2)$$

We will use this approach throughout the formulation.

The flow constraints are classical “flow in equals flow out” constraints. The flow constraints for vertices associated with inbound stations are trivial, as arriving trains immediately begin their dwell time. This leads to the following equation:

$$x_{s',t-\epsilon_{s'}}^l = d_{s,t}^l \quad (s', s) \in E^l, \quad t \in T, \quad \forall l \in L. \quad (3)$$

Note that equation (2) ensures that in the above equation those variables with zero and negative indices are zero.

For outgoing station vertices, note that there are only two incoming arcs; those resulting from trains finishing their dwell times and those from trains that waited in the previous time period. Similarly, there are only two outgoing arcs, one representing waiting and one representing leaving the station. This leads to the equality:

$$d_{s,t-D_s^l}^l + w_{s,t-1}^l = w_{s,t}^l + x_{s,t}^l \quad s \in S^l \setminus \omega, \quad t \in T, \quad \forall l \in L. \quad (4)$$

The last flow-type inequality ensures that there is exactly one unit of flow for each train throughout the track. While it is sufficient to ensure this by the constraint  $\sum_{t \in T} x_{\alpha,t}^l = 1$  for all  $l \in L$ , we include redundant (weaker) constraints representing the flow out of each station.

$$\sum_{t \in T} x_{\alpha,t}^l = 1 \quad \forall l \in L \quad (5)$$

$$\sum_{t \in T} x_{s,t}^l \leq 1 \quad \forall s \in S^l \setminus \{\alpha, \omega\}, \quad \forall l \in L. \quad (6)$$

For notational purposes, we write the above flow constraints (3)-(6) as  $A_F [x \ d \ w]^\top \leq b_F$ . Let  $P_F$  be the polytope defined by  $P_F = \{(x, d, w) \in [0, 1]^n : A_F [x \ d \ w]^\top \leq b_F\}$ , where  $n$  is the dimension of the problem. Note that  $A_F$  can be written as a block diagonal matrix, where block  $l$  represents the flow constraints of train  $l$ . Since each of these blocks are totally unimodular matrices,  $A_F$  is totally unimodular, meaning that  $P_F$  is an integer polytope.

The last set of constraints, labeled track-occupation constraints, ensure that there is at most one train on a given track at any given time. The track-occupation constraints are modeled by the set packing constraints:

$$\sum_{\{l \in L | (s, s') \in E^l\}} \sum_{\{t' \in T | t \in [t', t' + e_s^l - 1]\}} x_{s, t'}^l + \sum_{\{l \in L | (s', s) \in E^l\}} \sum_{\{t' \in T | t \in [t', t' + e_{s'}^l - 1]\}} x_{s', t'}^l \leq 1 \quad \forall t \in T, \forall (s, s') \in S \times S. \quad (7)$$

Note that the first sum keeps track of trains leaving  $s$  and going toward  $s'$  and the second sum keeps track of trains leaving  $s'$  and going toward  $s$ . We let  $A_O x \leq \mathbf{1}$  represent the track-occupancy constraints (7), and let  $P_O = \{(x, d, w) \in [0, 1]^n : A_O x \leq \mathbf{1}\}$  be the polytope associated with the track-occupancy constraints (7). Note that as written,  $P_O$  and  $P_F$  have the same dimension. We will sometimes refer to the projection of  $P_O$  onto the  $x$  variables,  $P_{O_X}$ , formed by truncating the  $d$  and  $w$  variables. Note that  $A_O$  can be written as a block diagonal matrix where each block represents the packing constraints for an individual section of track. While linear relaxations of set packing formulations are not integral in general, these particular constraints for a given track can be interpreted as clique constraints on an interval graph, which have a totally unimodular structure (Nemhauser and Wolsey 1988). Since  $A_O$  can be written as a block diagonal matrix where each block is totally unimodular, it is totally unimodular, thus integer.

As an example of the interval graph structure, let Figure 2 represent the section track connecting stations two and three. There are two trains. The first, train one, is traveling from station two to station three and can leave at any time between sixth and tenth hours. Train two, traveling from station three to station two, can leave station three between the hours of six and nine. We assume travel time for both trains is three hours. Figure 2 denotes each potential travel assignment and displays when each of the trains is on the track for all assignments. Track occupancy constraints for each hour are formed by summing up the variables that overlap that particular hour in the

interval graph. The nontrivial occupancy constraints for this example are:

$$\begin{aligned}
x_{2,6}^1 + & & x_{3,6}^2 & \leq 1 \\
x_{2,6}^1 + x_{2,7}^1 + & & x_{3,6}^2 + x_{3,7}^2 & \leq 1 \\
x_{2,6}^1 + x_{2,7}^1 + x_{2,8}^1 + & & x_{3,6}^2 + x_{3,7}^2 + x_{3,8}^2 & \leq 1 \\
& x_{2,7}^1 + x_{2,8}^1 + x_{2,9}^1 + & & x_{3,7}^2 + x_{3,8}^2 + x_{3,9}^2 \leq 1 \\
& & x_{2,8}^1 + x_{2,9}^1 + x_{2,10}^1 + & & x_{3,8}^2 + x_{3,9}^2 \leq 1 \\
& & & x_{2,9}^1 + x_{2,10}^1 + & & x_{3,9}^2 \leq 1 \\
& & & & x_{2,10}^1 & \leq 1.
\end{aligned}$$

As Figure 2 is an interval graph and the above constraints are clique constraints, the above system of equations is totally unimodular.

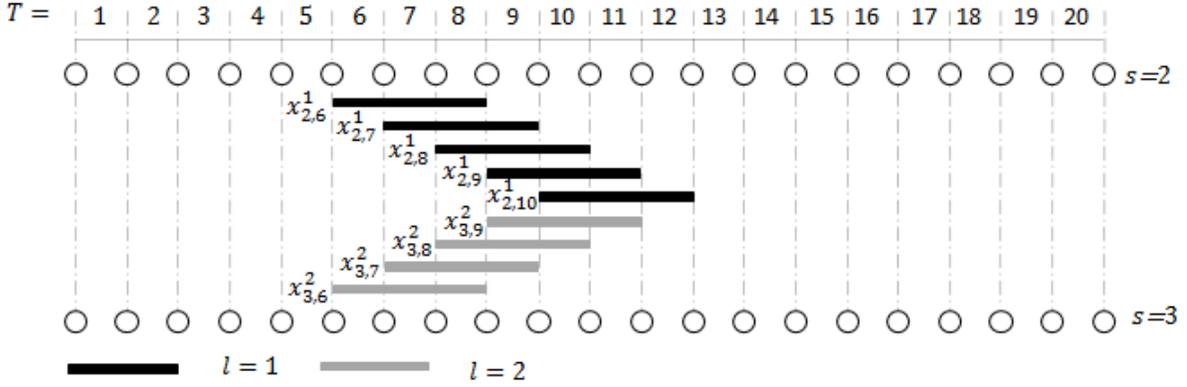


Figure 2 Track Occupancy as an Interval Graph

Note that the proposed formulation ignores headway constraints. This will be discussed in Section 3.2

### 3. Polyhedral Study of the Model

Unfortunately, the intersection of  $P_O$  and  $P_F$  does not always yield an integer polyhedron. The reason for this is as follows.  $P_O$  only ensures that two trains cannot be on the same track at the same time. Some solutions in  $P_O$  may have the same train assigned to the same track in two different time periods. These solutions are made infeasible in  $P_O \cap P_F$  by constraints (6). Unfortunately, adding constraints (6) to the formulation for  $P_O$  may break the interval graph structure, and thus breaking the polytope's integrality. Indeed, as  $A_F$  is block decomposable by trains and  $A_O$  is block decomposable by tracks, it is unlikely that  $[A_O \ A_F]^T$  has any (nontrivial) block diagonal structure. This can be seen in Figure 2. Flow constraints imply the constraints  $\sum_{i=6}^{10} x_{2,i}^1 = 1$  and  $\sum_{i=6}^9 x_{3,i}^2 = 1$ . These constraints break the interval graph structure of the track occupancy constraints and thus

break the integrality of the resulting polytope. For example, the polyhedron  $P_O \cap P_F$  associated with the problem described in Figure 2 contains one fractional vertex (defined by  $x_{2,7}^1 = x_{2,10}^1 = x_{3,6}^2 = x_{3,9}^2 = \frac{1}{2}$ ).

Suppose, however, that the flow clique constraints (6) in  $P_F$  were redundant to  $P_O$ . This will happen if and only if for each train  $l$  and each station  $s \in S^l$ , the difference between the latest possible departure and the earliest possible departure is less than the travel time, that is,  $\bar{T}_s^l - \underline{T}_s^l < \epsilon_s^l$ . Note that this condition implies that train  $l$  will be traveling somewhere on the track at time  $\bar{T}_s^l$ . Under this condition, with a minor assumption on the  $\rho$  parameters, we can show that the linear programming relaxation to the train scheduling problem will have an integer-optimal solution.

Let  $P_{LP}$  denote the linear relaxation of the train scheduling model, that is,  $P_{LP} = P_O \cap P_F$ .

**Condition 1**  $\bar{T}_s^l - \underline{T}_s^l < \epsilon_s^l$  for all  $l \in L$  and  $s \in S^l$ .

**Condition 2**  $\rho_{s,t}^l \geq \rho_{s,t+1}^l \geq 0$  and  $\mu_{s,t}^l \geq 0$  for all  $s \in S$ ,  $t \in T$ , and  $l \in L$ .

Note that Condition 2 not holding would imply that there are times at which a train's delayed leaving a station is preferable. In addition, it would imply that (more) waiting at stations is preferable.

**THEOREM 1.** *There exists an integer-optimal solution to the problem of maximizing the train scheduling problem over the polytope  $P_{LP}$  if Condition 1 and Condition 2 are met.*

*Proof of Theorem 1.* In order to prove Theorem 1, we first prove that a relaxed polytope defined by

$$P_R = \{(x, d, w) \in [0, 1]^{n_1} \times [0, 1]^{n_2} \times [-1, 1]^{n_3} : A_F \begin{bmatrix} x \\ d \\ w \end{bmatrix} \leq b_F, A_O x \leq \mathbf{1}\},$$

is integral where  $n_1$ ,  $n_2$ , and  $n_3$  are the sizes of  $x$ ,  $d$ , and  $w$  variables respectively. Note that  $P_R$  is formed by relaxing the non-negativity constraints on the  $w$  variables.

First, note that Condition 1 implies that for every station  $s$  and every train  $l$ , train  $l$  is somewhere on the track departing  $s$  at time  $\bar{T}_s^l$ . The implication from constraint (7) is that we can fix variables representing other train's use of the track at time  $\bar{T}_s^l$  to zero, i.e.,

$$x_{s, \bar{T}_s^l}^{l'} = 0 \quad \forall s \in S, \text{ s.t. } (s, s') \in E^l, (s, s') \in E^{l'}, \forall l, l' \in L, l \neq l', \quad (8)$$

and

$$x_{s', \bar{T}_s^l}^{l'} = 0 \quad \forall s' \in S, \text{ s.t. } (s, s') \in E^l, (s', s) \in E^{l'}, \forall l, l' \in L, l \neq l'. \quad (9)$$

Note that the fixed variables must take the value of zero in  $P_O \cap P_F$  as well.

As a result of these fixings, the track-occupancy constraints (7) at time  $\bar{T}_s^l$  are:

$$\sum_{t' \in [\underline{T}_s^l, \bar{T}_s^l]} x_{s,t'}^l \leq 1 \quad \forall s \in S^l \setminus \{\alpha, \omega\}. \quad (10)$$

Since we know that train  $l$  must be on the track at time  $\bar{T}_s^l$ , we know that the above constraint must be binding, that is:

$$\sum_{t' \in [\underline{T}_s^l, \bar{T}_s^l]} x_{s,t'}^l = 1 \quad \forall s \in S^l \setminus \{\alpha, \omega\}. \quad (11)$$

Note that the equalities (8)-(11) describe a face of  $P_{O_X}$ , implying that

$$P_{O_X}^1 = P_{O_X} \cap \{x \mid x \text{ satisfies (8) - (11)}\} \quad (12)$$

is integral.

We now show that  $P_R$  is integral by identifying a one-to-one mapping from  $P_R$  to  $P_{O_X}^1$  and another one-to-one mapping from  $P_{O_X}^1$  to  $P_R$ . The former is trivial as truncating the  $d$  and  $w$  variables of a point in  $P_R$  will yield a unique point in  $P_{O_X}^1$ . A mapping from  $P_{O_X}^1$  to  $P_R$  can be identified by rewriting the flow equalities (4) as follows:

$$w_{s,t}^l = \sum_{\underline{T}_{s'}^l \leq t' \leq t - D_s^l} x_{s',t'}^l - \sum_{\underline{T}_s^l \leq t' \leq t} x_{s,t'}^l \quad \forall l \in L, (s', s) \in E^l, t < \bar{T}_s^l. \quad (13)$$

$$\sum_{\underline{T}_{s'}^l \leq t' \leq \bar{T}_{s'}^l} x_{s',t'}^l - \sum_{\underline{T}_s^l \leq t' \leq \bar{T}_s^l} x_{s,t'}^l = 0 \quad \forall l \in L, (s', s) \in E^l. \quad (14)$$

Note that constraint (14) is implied by (11) (since both of the sums add up to one), so it is redundant. Thus, any point in  $P_{O_X}^1$  can be mapped to a point in  $P_R$  by preserving the  $x$  values and using (13) to determine the  $w$  values. The  $d$  values can be identified using (3).

Note that the above one-to-one mappings transform integer points from  $P_{O_X}^1$  to  $P_R$ , and vice versa, so that  $-1 \leq w \leq 1$  does hold in  $P_R$ . Now we show that if optimizing over  $P_R$  produces a solution that is not in  $P_{LP}$ , then there is corresponding variable  $x_{s,t}^{l*}$  such that  $x_{s,t}^{l*} = 0$  is a valid cut for  $P_{LP}$  and that  $P_R$  maintains integrality after fixing  $x_{s,t}^{l*}$  to zero. The integrality stems from the fact that  $x_{s,t}^{l*} = 0$  gives a face of  $P_R$ , which we know to be integer. The fact that such an  $x_{s,t}^{l*}$  exists comes from identifying properties of the optimal solutions.

CLAIM 1. *Fixing  $x_{s,\underline{T}_s^l}^l$  to one for all trains  $l$  and stations  $s$  will, along with the lifting equalities (13) and (3), produces an optimal solution to the problem of maximizing (1) over  $P_R$ .*

The proof of this claim can be easily seen by recalling that Condition (2) implies that earlier departures are always preferable to late departures. Assuming proper preprocessing is performed (notably (8) & (9)), this solution will be feasible in  $P_R$  so long as  $P_R$  is nonempty.

If the solution found when optimizing over  $P_R$  does not have a negative  $w$  values, then it is an integer-optimal solution to the train scheduling problem. Suppose that there is some negative  $w_{s,t}^l$  (with the value of  $-1$ ). The fact that  $w_{s,t}^l = -1$  implies that train  $l$  arrives station  $s$  *after* time  $t$ , but leaves  $s$  *before* time  $t$ . As a result of Claim 1, however, the train could not have left its previous station earlier than what is currently being assigned, meaning that there is no feasible flow in  $P_F$  that can route a nonzero amount of flow along the corresponding arc. In other words, there is no solution in  $P_F$  (likewise in  $P_{LP}$ ) where  $x_{s,t}^l$  is not zero. Fixing  $x_{s,t}^l$  to zero maintains the integrality of  $P_R$  (since it defines a face) while also removing this solution from its feasible region. Iteratively optimizing over  $P_R$  will eventually produce an integer (vertex) solution that is feasible in  $P_O \cap P_F$ . As  $P_R$  is a relaxation, this solution is optimal to the problem of maximizing (1) over  $P_{LP}$ .  $\square$

### 3.1. Decomposing the $P_{LP}$ Region into Integer Polytopes

Theorem 1 is not surprising considering how easy the scheduling problem will be when Condition 1 is met, as it implies an order on when trains traverse a given section of track. While Condition 1 is not likely to hold in many real-world instances, the intuition can be used to effectively decompose  $P_{LP}$  into integer polytopes by creating subproblems where Condition 1 does hold. In this section, we create an extended formulation for the train scheduling problem by introducing new variables that represent windows for when trains traverse each track. By ensuring that each of these windows satisfies Condition 1, we can decompose  $P_{LP}$  into integer polytopes.

For a given train  $l$  and station  $s$ , we partition the interval  $[\underline{T}_s^l, \overline{T}_s^l]$  into smaller intervals by letting  $I_{s,j}^l$  represent the interval:

$$I_{s,j}^l = [\underline{T}_s^l + j\epsilon_s^l, \underline{T}_s^l + (j+1)\epsilon_s^l - 1]. \quad (15)$$

Let  $k_s^l$  denote the number of partitioned intervals for train  $l$  at station  $s$ , i.e.,  $k_s^l = \lceil (\overline{T}_s^l - \underline{T}_s^l + 1)/\epsilon_s^l + 1 \rceil$ . For any pair of partitioned intervals for train  $l$  at station  $s$ , we have  $[\underline{T}_s^l, \overline{T}_s^l] \subset \cup_{j=0}^{k_s^l} I_{s,j}^l$  and  $I_{s,j}^l \cap I_{s,j'}^l = \emptyset$  when  $j \neq j'$ . We now let  $z_{s,j}^l \in \{0, 1\}$  represent train  $l$  leaving station  $s$  in interval  $I_{s,j}^l$ , that is:

$$z_{s,j}^l = \sum_{t \in I_{s,j}^l} x_{s,t}^l \quad \forall l \in L, s \in S^l, 0 \leq j \leq k_s^l. \quad (16)$$

Now consider a solution to following extended formulation:

$$\max \sum_{l \in L} \sum_{s \in S^l} \sum_{t \in T} \rho_{s,t}^l x_{s,t}^l - \mu_{s,t}^l w_{s,t}^l \quad (17)$$

$$\text{Subject To: } x_{s,t}^l = 0 \quad \forall l \in L, s \in S^l \setminus \omega \text{ and } t \notin [\underline{T}_s^l, \overline{T}_s^l] \quad (18)$$

$$x_{s',t-\epsilon_{s'}}^l = d_{s,t}^l \quad \forall l \in L, (s',s) \in E^l, t \in T \quad (19)$$

$$d_{s,t-D_s^l}^l + w_{s,t-1}^l = w_{s,t}^l + x_{s,t}^l \quad \forall l \in L, s \in S^l \setminus \omega, t \in T \quad (20)$$

$$\sum_{t \in T} x_{s,t}^l = 1 \quad \forall l \in L, \forall s \in S^l \setminus \omega \quad (21)$$

$$\begin{aligned} & \sum_{\{l \in L | (s,s') \in E^l\}} \sum_{\{t' \in T : t \in [t', t' + \epsilon_s^l - 1]\}} x_{s,t'}^l + \\ & \sum_{\{l \in L | (s',s) \in E^l\}} \sum_{\{t' \in T : t \in [t', t' + \epsilon_{s'}^l - 1]\}} x_{s',t'}^l \leq 1 \quad \forall t \in T, s, s' \in S \end{aligned} \quad (22)$$

$$z_{s,j}^l = \sum_{t \in I_j^l} x_{s,t}^l \quad \forall l \in L, s \in S^l, 0 \leq j \leq k_s^l \quad (23)$$

$$z_{s,j}^l \in \{0, 1\} \quad \forall l \in L, s \in S^l, 0 \leq j \leq k_s^l \quad (24)$$

$$x \in [0, 1]^{n_1} \quad (25)$$

$$w \in [0, 1]^{n_2} \quad (26)$$

$$d \in [0, 1]^{n_3}. \quad (27)$$

Even though the integer restrictions on the  $x$ ,  $w$ , and  $d$  variables are relaxed, there is an optimal solution to the extended formulation that has integer  $x$ ,  $w$ , and  $d$  variables.

**COROLLARY 1.** *There exists an optimal solution to the extended formulation contains integer  $x$ ,  $w$ , and  $d$  variables.*

*Proof of Corollary 1.* The intervals  $I_{s,j}^l$  are chosen to guarantee that they satisfy Condition 1. Thus, Theorem 1 applies to the subproblem formed by fixing all of the  $z$  variables to integer values.  $\square$

Note that the above model contains far fewer integer variables than the original model, as there will be  $\lceil (\overline{T}_s^l - \underline{T}_s^l + 1) / \epsilon_s^l + 1 \rceil$  many  $z$  variables associated with train  $l$  at station  $s$ , compared to the  $(\overline{T}_s^l - \underline{T}_s^l + 1)$ -many  $x$  variables. Also note that the number of  $z$  variables does not increase if finer time increments are used as  $\overline{T}_s^l$ ,  $\underline{T}_s^l$ , and  $\epsilon_s^l$  all scale proportionally. So, while finer time increments (minutes opposed to hours for instance), will lead to larger problem instances, the size of the branch-and-bound tree is likely to remain unchanged.

### 3.2. Extensions to the Model

#### *Headway Constraints:*

The above formulation ensures that only one train can be on a track at a time. This is necessary if two trains are traveling in the opposite direction. However, some train scheduling problems may allow for trains to be on the same track at the same time provided that they are traveling in the same direction and there is a sufficiently large distance between them. Constraints that allow for some overlap, restrictions of the constraints in (7), are known as headway constraints. Headway constraints are typically modeled by identifying *conflicting* assignments. That is, if assigning train  $l$  to depart station  $s$ , toward  $s'$ , at time  $t$  while also assigning train  $l'$  to depart station  $s$ , toward  $s'$ , at time  $t'$  will cause a violation on the minimum distance between the trains, then the headway constraint  $x_{s,t}^l + x_{s,t'}^{l'} \leq 1$  is added to the problem formulation. Note that these constraints are a function of each train's speed and directions. Adding the constraints for all conflicting assignments will break the interval graph structure of the occupancy constraints, thus potentially breaking the integrality of  $P_O$ . The natural algorithmic approach to deal with these conflict-type constraints is to generate all of the clique inequalities that arise from the conflict-type constraints (since the conflict-type constraints do in essence define an independent set on a graph). As the conflict-type constraints are dominated by the clique inequalities, they can be removed from the formulation (Caimi et al. 2011, Caprara et al. 2002). Even with the clique inequalities, the resulting description of  $P_O$  is not likely, in general, to be integral. However, the structure imposed by Condition 1 is enough to guarantee the integrality of  $P_O$ .

Recall that Condition 1 implies that the sequence of trains traveling along a given track is known a priori. In addition, we know times where each train has to travel on each track. More precisely, for each train  $l$  and station  $s$ , we know that  $l$  is on the track departing  $s$  at time  $\bar{T}_s^l$ . Suppose train  $l'$  is the next train to use the track after  $l$ . If trains  $l$  and  $l'$  are traveling in opposite directions, then the track occupancy constraints for times in the interval  $[\bar{T}_s^l + 1, \bar{T}_{s'}^{l'} - 1]$  is:

$$\sum_{\{t' \in T | t \in [t', t' + e_s^l - 1]\}} x_{s,t'}^l + \sum_{\{t' \in T | t \in [t', t' + e_s^{l'} - 1]\}} x_{s,t'}^{l'} \leq 1 \quad \forall t \in [\bar{T}_s^l + 1, \bar{T}_{s'}^{l'} - 1]. \quad (28)$$

Note that the above set of constraints are just those from (7), so they maintain the interval graph structure. Now, suppose train  $l$  and  $l'$  are traveling in the same direction. Let  $h_{s,s'}$  denote the minimum time required between two trains leaving  $s$  toward  $s'$ . The headway constraints can be written as:

$$\sum_{\{t' \in T | t \in [t', t' + h_{s,s'}^l - 1]\}} x_{s,t'}^l + \sum_{\{t' \in T | t \in [t', t' + e_s^{l'} - 1]\}} x_{s,t'}^{l'} \leq 1 \quad \forall t \in [\underline{T}_s^{l'}, \bar{T}_{s'}^{l'} - 1], \quad (29)$$

which still maintains the interval graph structure (since (29) is equivalent to (7) when  $\epsilon_s^l$  is replaced by  $h_{s,s'}$ ). As a result, the Theorem 1 still applies.

There may be other conflict-type constraints introduced by other physical constraints. For instance, *arrival time* constraints may limit the number of trains that arrive at a station in a given interval while *departure time* constraints may bound the number of trains that leave a station during a given interval. These types of constraints can overlap different sections of tracks (for instance, train  $l$  cannot leave  $s$  going northbound at the same time that  $l'$  leaves  $s$  going southbound), breaking the decomposability of  $P_O$ , and thus its integrality.

*Dropping Trains:*

In some train scheduling problems, decision makers have the option of removing trains from the system (if, for instance, it is infeasible to route all of the trains). The polyhedral results discussed in this paper only deal with situation where each train has to be scheduled. However, the possibility of dropping trains can be added to the extended formulations by replacing constraints (21) with:

$$\sum_{t \in T} x_{s,t}^l \leq 1 \quad \forall l \in L, \forall s \in S^l \setminus \omega.$$

The above constraint allows for a zero flow throughout the network for a given train. With no flow, the  $z$  variables associated with that train will take the value of zero in the extended formulation. As a result, the polyhedron formed by fixing  $z$  variables to binary values will still be integer optimal (if feasible).

*Variable-Speed Trains:*

The model discussed in this paper assumes that each train can travel at a given speed along a given section of track. There may be cases where a train has the option of traveling at different speeds. Unfortunately, the claim made in the proof to Theorem 1 does not necessarily hold if trains can move at variable speeds. However, while we prove that the polytope discussed in Theorem 1 is integer optimal, our intuition leads us to believe that the polytope might be integer. If so, partitioning the travel arcs, even those representing different speeds, would result in an integer polytope so long as for each position, there is at least one time period in common with every arc in that partition.

## 4. Computational Experiments

In this section, we present the computational results of our method on some real-world instances provided by the Iranian Railway Company. First, we briefly describe the railway infrastructure and the parameters.

The computational instances include a railway path in which consecutive stations are connected via a single track. The number of trains and stations are different in the instances. Each train

has an origin and destination that are aligned with a path that may be different for other trains. Trains are classified as either Express or Regular, where the classification affects the trains' travel speed and profit. A train gets the highest profit if it arrives its destination at earliest possible arrival time. The profits corresponding to Express trains are greater than Regular. Note that train's profit would be reduced proportional to waiting times at stations that are caused by operational constraints, specified in our model assumptions. Moreover, for instances 1 to 16, each train has 60-minute dispatching ticket, whereas 120-minute dispatching ticket for each train of instances 17 to 31. That is, a train departs its origin within 60 (120) minutes. Note that trains' dispatching tickets are distributed across the scheduling time horizon. Finally, the scheduling time horizon is set to 48 hours. All computational instances are available online (see Barah et al. (2017) in References).

We solved all computational instances using IBM CPLEX 12.7.1 (64 bit edition) on a computer running MS WINDOWS 10 (64 bit edition) with two INTEL XEON, 2.4 GHz processors and 64GB of RAM. Note that the computational time was limited to 3,600 seconds for each instance. For the instances that were not solved in the given time limit, we report the optimality gap at 3,600 seconds.

Table 4 shows the performance of original model, described in constraints (1)-(7) and the extended formulation, described in constraints (17)-(27). Two separate batches of tests were run. The first batch was formed by discretizing the time horizon into 5-minute intervals. Using the granular discretization, CPLEX was able to solve 30 out of 31 instances of the original formulation and all instances of the extended formulation. As the results indicate, CPLEX was able to solve the extended formulation in about half of the time required to solve the original formulation. The reason for this is likely due to the fewer number of binary variables in the formulation. Table 4 gives the number of binary variables representing travel arcs ( $x$  variables) in the original formulation under the column title "Arcs," as there is a variable for every arc in the graph while "Binaries" denotes the number of binary variables in the extended formulation ( $z$  variables). As seen in the table, the extended formulation contains about one third as many binary variables as the original. The difference between the two models is more pronounced when we discretize the time horizon into 1-minute intervals. CPLEX was only able to solve 17 of the original problems instances and 25 of the extended formulation instances to optimality in the given time limit. For those instances that CPLEX failed to solve them to optimality, we reported the gap, in percentage. Table 4 gives the average time for solving each of these instances including the unsolved instances (where times are assumed to be 3,600 seconds). In this case, using the extended formulation is shown to be 45% faster. Moreover, as it is shown in the last row of the table, if we restrict ourselves to instances where CPLEX was able to solve both formulations in the time limit, we see that the extended formulation is approximately three times faster given 1-minute intervals (approximately two times

faster given 5-minute intervals). An explanation for the drastic differences in solution times can be seen by looking at the number of binary variables in each of the formulations. As discussed in Section 3.1, the size of the extended formulation is not sensitive to time discretization. The number of binary variables only changes slightly (the differences are due to the rounding down of the travel times during the discretization). The number of binary variables in the original formulation, however, grows by a factor of 5. As a result, there is an order of magnitude fewer binary variables in the extended formulation than in the original formulation.

Another explanation for the dramatic decrease in solution times is as follows. In the original model, branching is done on the individual  $x$  variables. While fixing an  $x$  variable to one is likely to produce a strong bound, fixing an  $x$  variable to zero is likely to result in no change in the corresponding LP value. Indeed, ensuring that train  $l$  does not leave station  $s$  at time  $t$  is not a strong requirement if it may instead leave a minute later. Introducing the  $z$  variables allows for more generalized branching disjunctions that serves to balance the strength of each disjunction, as fixing  $z$  to either zero or one is likely to have a big impact on the LP bound. Another reason is that there are likely many similar optimal and near optimal solutions to the train scheduling problem, as delaying a random train by one minute is likely to maintain feasibility without a significant impact on the objective function. The presence of many optimal and near-optimal solutions tends to slow down branch-and-bound solvers, as pruning nodes become more difficult if they contain optimal solutions. Branching on the  $z$  variables helps cluster these optimal and near-optimal solutions into a smaller number of branch-and-bound nodes.

Another metric of interest is how different formulations impact the time to find a (provably) near-optimal solution. To test this, we reran the computational experiments using an optimality tolerance of 1%. Using a coarser gap, we would expect that root node heuristics and cuts had a larger impact on the overall solution process than branching did. Such intuition was shown to be valid, as the difference between the two formulations were much less dramatic. Given the 1-minute time intervals, the average solution time (excluding instance number 29) for the original formulation was 701 seconds compared to 661 seconds for the extended formulation. With a coarser gap, the extended formulation only gives a 6% speedup.

## 5. Conclusion

We have formulated an extended formulation to train scheduling problems with drastically fewer binary variables compared to the basic IP model. Branching on the newly formed variables allows us to decompose the problem into integer-optimal subproblems. Our computational results show that this new approach can significantly outperform the original IP model in terms of solution time, the number of instances solved to optimality, and the relative optimality gap within instances where

both models found feasible solutions. Moreover, this approach allows us to solve train scheduling problems at much finer time discretizations without drastically increasing the solution time.

Instance Data			5-Minute Intervals				1-Minute Intervals			
Number	$ L $	$ S $	Original Formulation		Extended		Original Formulation		Extended	
			Arcs	Time (Nodes) or (Gap %)	Binaries	Time (Nodes)	Arcs	Time (Nodes) or (Gap %)	Binaries	Time (Nodes) or (Gap %)
1	15	47	6,656	5 (13)	2,228	5 (123)	31,232	147 (985)	2,256	115 (93)
2	15	47	6,448	8 (3,090)	2,149	7 (1,196)	30,256	85 (727)	2,176	47 (99)
3	15	47	7,007	70 (10,051)	2,306	17 (1,410)	32,879	272 (1,193)	2,333	153 (511)
4	15	47	6,448	6 (678)	2,162	4 (570)	30,256	30 (981)	2,191	20 (33)
5	15	47	6,487	16 (2,884)	2,157	12 (2,256)	30,439	46 (305)	2,183	38 (29)
6	20	47	8,034	67 (11,570)	2,657	17 (1,830)	37,698	40 (29)	2,689	59 (75)
7	20	47	8,398	12 (432)	2,796	11 (825)	39,406	2,607 (12,193)	2,833	691 (1,195)
8	20	47	8,203	333 (37,537)	2,710	101 (13,542)	38,491	266 (1,236)	2,742	161 (292)
9	20	47	7,787	5 (0)	2,608	7 (109)	36,539	(0.23 %)	2,642	569 (696)
10	20	47	7,930	74 (11,853)	2,632	20 (2,720)	37,210	(0.21 %)	2,664	332 (1,113)
11	25	47	8,190	134 (18,573)	2,680	69 (8,770)	38,430	384 (8,132)	2,709	119 (804)
12	25	47	9,035	523 (41,110)	2,993	32 (3,625)	42,395	(0.17 %)	3,029	778 (1,295)
13	25	47	8,970	145 (10,354)	2,951	122 (8,202)	42,090	181 (327)	2,985	150 (108)
14	25	47	9,230	202 (15,304)	3,038	79 (7,014)	43,310	(0.56 %)	3,073	1,681 (1,495)
15	25	47	8,242	475 (54,185)	2,736	74 (9,611)	38,674	23 (0)	2,769	13 (0)
16	30	47	10,270	163 (11,522)	3,405	128 (11,119)	48,190	(0.38 %)	3,446	1,966 (1,543)
17	15	52	15,725	6 (0)	6,153	8 (17)	76,109	(0.04 %)	6,940	831 (305)
18	15	52	14,200	5 (9)	5,532	4 (36)	68,728	22 (0)	6,241	16 (3)
19	15	52	14,150	15 (539)	5,530	13 (169)	68,486	3,581 (8,343)	6,239	625 (240)
20	15	52	15,950	10 (11)	6,302	8 (54)	77,198	49 (0)	7,098	38 (3)
21	15	52	14,975	9 (29)	6,007	8 (98)	72,479	102 (0)	6,775	82 (3)
22	20	52	18,750	3 (0)	7,246	4 (0)	90,750	57 (0)	8,159	39 (3)
23	20	52	19,000	17 (298)	7,388	13 (219)	91,960	(0.09 %)	8,293	1,447 (387)
24	20	52	20,325	1,269 (25,733)	7,963	1,270 (20,641)	98,373	(0.3 %)	8,963	(0.36 %)
25	20	52	19,950	15 (263)	7,686	15 (211)	96,558	(0.09 %)	8,637	335 (1,234)
26	20	52	18,975	13 (152)	7,321	16 (601)	91,839	85 (9)	8,253	65 (39)
27	25	52	24,675	(0.03 %)	9,727	1,502 (19,119)	111,199	(0.42 %)	10,197	(0.35 %)
28	25	52	22,975	333 (14,029)	9,059	436 (9,988)	115,192	(0.4 %)	10,479	(0.51 %)
29	25	52	23,800	2,968 (63,988)	9,296	352 (6,223)	120,032	(10.62 %)	10,909	(3.88 %)
30	25	52	24,800	1,080 (17,229)	9,684	1,298 (14,128)	115,918	(0.44 %)	10,708	(0.32 %)
31	25	52	23,950	258 (12,003)	9,470	69 (4,341)	119,427	(0.69 %)	10,967	(0.55 %)
Solved Instances			30		31		17		25	
Average (all)			> 382		185		> 1,884		> 1,032	
Average (both solved)			274		141		469		143	

**Table 4** Computational comparison between original and extended formulations. Given a discretized time interval, the last row of the table shows the average solution time (in seconds) restricted to instances that both formulations solved them to optimality.

## Acknowledgments

J. Ostrowski was supported by NSF award number 1332662 and DOE (ASCR) award DE-SC0018175.

## References

- Barah, Masoud, Abbas Seifi, James Ostrowski. 2017. Train scheduling problems. [https://github.com/mbarah/Train\\_Scheduling\\_Problems](https://github.com/mbarah/Train_Scheduling_Problems).
- Borndörfer, Ralf, Thomas Schlechte. 2007. 05. Models for Railway Track Allocation. Christian Liebchen, Ravindra K. Ahuja, Juan A. Mesa, eds., *7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'07), OpenAccess Series in Informatics (OASICs)*, vol. 7. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany. doi:10.4230/OASICs.ATMOS.2007.1170. URL <http://drops.dagstuhl.de/opus/volltexte/2007/1170>.
- Brännlund, Ulf, Per Olov Lindberg, Andreas Nöu, J-E Nilsson. 1998. Railway timetabling using lagrangian relaxation. *Transportation science* **32**(4) 358–369.
- Cacchiani, Valentina, Alberto Caprara, Paolo Toth. 2008. A column generation approach to train timetabling on a corridor. *4OR* **6**(2) 125–142.
- Cacchiani, Valentina, Alberto Caprara, Paolo Toth. 2010a. Non-cyclic train timetabling and comparability graphs. *Operations Research Letters* **38**(3) 179–184.
- Cacchiani, Valentina, Alberto Caprara, Paolo Toth. 2010b. Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological* **44**(2) 215–231.
- Cacchiani, Valentina, Paolo Toth. 2012. Nominal and robust train timetabling problems. *European Journal of Operational Research* **219**(3) 727–737.
- Cai, Xiaoqiang, CJ Goh. 1994. A fast heuristic for the train scheduling problem. *Computers & Operations Research* **21**(5) 499–510.
- Caimi, Gabrio, F Chudak, Martin Fuchsberger, Marco Laumanns, Rico Zenklusen. 2011. A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling. *Transportation science* **45**(2) 212–227.
- Caimi, Gabrio, Frank Fischer, Thomas Schlechte. 2018. *Railway Track Allocation*. Springer International Publishing, Cham, 141–159. doi:10.1007/978-3-319-72153-8\_7. URL [https://doi.org/10.1007/978-3-319-72153-8\\_7](https://doi.org/10.1007/978-3-319-72153-8_7).
- Caprara, Alberto, Matteo Fischetti, Paolo Toth. 2002. Modeling and solving the train timetabling problem. *Operations research* **50**(5) 851–861.
- Caprara, Alberto, Michele Monaci, Paolo Toth, Pier Luigi Guida. 2006. A lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete applied mathematics* **154**(5) 738–753.
- Carey, Malachy, David Lockwood. 1995. A model, algorithms and strategy for train pathing. *Journal of the Operational Research Society* 988–1005.

- Erol, Berkan. 2009. Models for the train timetabling problem. Ph.D. thesis, Technischen Universitt Berlin.
- Harrod, Steven, Thomas Schlechte. 2013. A direct comparison of physical block occupancy versus timed block occupancy in train timetabling formulations. *Transportation Research Part E: Logistics and Transportation Review* **54** 50 – 66. doi:<https://doi.org/10.1016/j.tre.2013.04.003>. URL <http://www.sciencedirect.com/science/article/pii/S136655451300063X>.
- Higgins, Andrew, Erhan Kozan, Luis Ferreira. 1997. Heuristic techniques for single line train scheduling. *Journal of Heuristics* **3**(1) 43–62.
- Lusby, Richard M. 2008. Optimization methods for routing trains through railway junctions. Ph.D. thesis, University of Auckland.
- Lusby, Richard M, Jesper Larsen, Matthias Ehrgott, David Ryan. 2011. Railway track allocation: models and methods. *OR spectrum* **33**(4) 843–883.
- Nemhauser, George L., Laurence A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, USA.
- Szpigel, B. 1973. Optimal train scheduling on a single track railway. Michael Ross, ed., *Proceedings of the sixth IFORS International Conference on Operational Research*. North-Holland Publishing Co., London, United Kingdom, 343–352.
- Velasquez, Rafael, Matthias Ehrgott, David Ryan, Anita Schöbel. 2005. A set-packing approach to routing trains through railway stations. *Proceedings of the 40th annual conference of the operational research society of New Zealand*. birthplace, New Zeland, 305–314.
- Zhou, Xuesong, Ming Zhong. 2007. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B: Methodological* **41**(3) 320–341.