Abstract

We consider the dynamic headway control problem under no-deadlock and no-collision constraints. Different speed limits are applied to different railway segments and junctions according to local geographic conditions. A mathematical programming formulation is given under the consideration of the trains' dynamics. Then a heuristic rule for train movement is proposed to reduce delay and guarantees a safety distance between trains. Simulation experiments are also conducted to show the efficiency of the dynamic headway model together with the proposed heuristics.

Keywords: Dynamic Headway; Scheduling; Simulation; Heuristics

1. Introduction

Railways, as one of the most cost-efficient ways to transport goods and people, serve as a major part of the transportation demand, especially for freight transportation in the United States. According to an Association of American Railways’ study [2], railway moves about 40% of freight measured in ton-miles, generating $71.6 billion of revenue in the United States in 2012. Based on the statistics given by the Federal Railroad Administration [16], the railway system will experience a 22% increase in the amount of tonnage from 2010 to 2035. However, it is very expensive to extend the current railway’s infrastructure. Therefore a better way of managing the railway system is needed.

One possible means to increase the efficiency of rail systems is through improved scheduling and dispatching. The train scheduling and dispatching problem can be classified into passenger train scheduling and freight train scheduling based on the train’s functionality. Passenger trains and freight trains differ due to their special features. Passenger train schedules are relatively stable and periodic. Meanwhile freight schedules are more flexible. Sometimes freight trains may even depart their origins without fixed schedules. Then real-time dispatching decisions are made during its travel towards its destination.

New communication technologies have the potential to improve railway operations, especially through more efficient train scheduling and dispatching. Positive Train Control (PTC) is introduced as a system of monitoring and controlling the movement of trains to increase security by reducing human operation (Badugu and Movva [3]). With PTC, trains can communicate with other trains to share information. Previously trains are ‘blind’ and controlled by the signals which are operated by experienced human dispatchers. With PTC, each train can have information of trains near it (‘locally’) and even trains far away from it (‘globally’).

In reality, trains are controlled by signals in order to avoid collision. Hence, the track segment between two consecutive signals works as headway between two consecutive trains. Therefore the railway track is typically modeled as a set of blocks, where each block can hold only one train at any time. As a consequence, headway between two consecutive trains is represented as a block with fixed length in most of the modelling approaches. The introduction of a PTC system enlarges the limited control given by the signals. The trains can be controlled to decelerate, accelerate and travel at a constant speed in real time at any point, resulting in dynamic headway between two consecutive trains. As a result existing modelling approaches using fixed headways cannot be used to represent dynamic headway control, especially when the dynamic headway contains only a fractional of one single block. Thus, we propose a new modelling framework for dynamic headway control rule taking the train’s dynamics into consideration.

Moreover, PTC technology provides finer monitoring and control over a train’s velocity. It leads to a better estimation of travel time over one segment. In contrast, most of the previous research focuses on constant velocity
when considering travel time over one segment. The introduction of a dynamic headway framework brings new problems that previous scheduling and routing models cannot deal with. One of the most challenging problems is routing trains with velocity control at each segment since the travel time over one segment is dependent on the entering velocity and exiting velocity. This type of velocity dependent routing is seldom considered in the previous research.

The purpose of this research is to further the state-of-the-art of the train scheduling and routing problem taking into consideration the new capabilities of dynamic headway control that the newly introduced technologies such as PTC provide. We note that most of the prior research on train scheduling assumes constant headway and often times referred to as fixed blocks and this paper to the best of our knowledge is one of the few papers that studies the train scheduling problem under dynamic headway. Specifically, the contribution of this research is (1) we develop a framework to represent dynamic headway and build a corresponding simulation model, and (2) we formulate an optimization model and propose a heuristic method to schedule trains in the new framework.

The rest of the paper is organized as follows. In section 2, the previous related literature is reviewed. A formal description of the problem is given in section 3 and due to the complexity of the optimization problem, we decompose the problem into headway decision, velocity decision and schedule and route decision. The headway decision and velocity decision are considered in section 4. Section 5 shows numerical results. Conclusions are drawn in Section 6.

2. Literature review

There have been several survey papers that focus on different aspects of the rail scheduling problem. Cordeau et al. [10] surveyed optimization models for the train routing and scheduling problem. Caprara et al. [7] gave a review of strategic, tactical and operational level’s decision models for passenger trains. Lusby et al. [23] reviewed different models and approaches for the train timetabling, train dispatching and train routing problem. This review was grouped into single track network models, general network models and junction routing models. Most recently, Harrod [17] surveys the train timetabling problem. This survey was organized by problem features (aperiodic or periodic, explicit track infrastructure modeling or not, etc.). Fang et al. [15] summarized models and methodologies for rescheduling in railway networks.

This paper touches on two areas where there has been prior research. Our work in developing a modeling framework for dynamic headway furthers the existing simulation approaches and the prior literature is reviewed in section 2.1. The prior work on the train scheduling problem is reviewed in section 2.2.

2.1. Railway simulation models

Even though much research has focused on using an optimization model to solve the train scheduling problem, the large number of integer variables makes it difficult to solve real problems in a relatively short time. Hence, researchers turn to simulation models to evaluate railway systems. The most difficult part in the simulation model is modeling the deadlock avoidance. Lu et al. [22] has shown that determining whether a state in the railway network will result in a deadlock is a NP-complete problem. Many approaches have been proposed to guarantee deadlock free movement for trains.

Lu et al. [22] proposed a detailed simulation approach that considered the trains’ dynamics. The authors formulated a graph representation for the track infrastructure for a complex network consisting of single-, double-, and triple-track lines. A greedy algorithm based heuristic was proposed to solve the train scheduling problem in the simulation model. Then the proposed approach was used to simulate the railway network in Los Angeles County.

Dorfman and Medanic [14] presented a discrete event model to simulate the railway network. A greedy travel advanced strategy (TAS) was presented and a capacity check approach was integrated to check the deadlock avoidance constraint. Then Li et al. [20] utilized global information about trains to improve the model of Dorfman and Medanic [14]. An effective travel advance strategy based (ETAS) on the global information about trains together with a deadlock avoidance control method was proposed in Li et al. [20]. A single lane railway network was simulated to test the new simulation model. Numerical results showed that ETAS had better performance than TAS.

Pachl [29] presented two approaches to avoid deadlock in the simulation model. Movement Consequence Analysis (MCA) achieved the goal by detecting the potential deadlock as a consequence of the train movement. In contrast, a Dynamic Route Reservation (DRR) reversed a number of blocks ahead of each train. Comparison was also made between MCA and DRR.

Marinov and Viegas [24] presented a decomposition approach in the simulation model. The entire railway system was separated into different components: rail lines, rail yards, rail stations, rail terminals and junctions. A queuing
system among the components was utilized to simulate the entire railway system. An event based simulation was proposed and implemented using the ARENA simulation software in Motraghi and Marinov [27]. Then two scenarios were evaluated by the simulation model. In the first one, both passenger and freight trains had fixed schedules while freight trains did not have a fixed timetable in the second scenario. The simulation model was used to evaluate the Metro rail system in Newcastle upon Tyne. Zhou and Mi [34] brought dynamics of train movements into the simulation model for a fixed-block railway network. Corresponding scheduling and control policies were also presented and experimental results showed that the speed performance is improved in the proposed model.

Wales and Marinov [32] developed an event based simulation system to evaluate the impact of the delay. Three delay mitigation tactics (to address primary delays, secondary delays and ahead of schedule running) are analyzed and compared. A similar event based simulation system is used in Abbott and Marinov [1] to help design railway interchange yards. They proposed a simulation method with progressive evaluations and revisions with the objective of integrating high speed railway lines with current conventional railways.

2.2. Train scheduling problems

One of the earliest works dates back to Szpigel [30], where the author related the train scheduling problem on a single track with the job shop scheduling problem. A branch and bound algorithm was proposed to solve this problem. Also experiments were conducted on a single track line with five track sections and ten trains.

Jovanovic and Harker [19] investigated the railway network with two stations with a lane connecting them and meet points along the lane, where trains can wait, overtake or be overtaken. Arrival and departure times at meet points together with a sequence of trains at meet points were introduced as decision variables to formulate a Mixed Integer Programming (MIP) problem.

Brannlund et al. [4] formulated a different model for the single track line scheduling problem by discretizing the time periods. Decision variables were indexed by train, track block and time period to indicate that the train occupied the track block at that time. Then a Lagrangian relaxation approach was used to solve the binary integer programming model. Experiments performed on the railway track in the middle of Sweden showed that near optimal schedules were obtained.

Both Jovanovic and Harker [19] and Brannlund [4] deal with a single track line railway system. Even though the railway infrastructure is simple, the two models proposed laid the foundation for problems with complex structure and formed two main streams of formulating the train scheduling problem. The main difference between them is that in the model originated from Jovanovic and Harker [19] times at control points are decision variables. While in the other stream, time is discretized to form a time-space graph. Much research has been done to modify, expand and improve these two basic models.

For the first stream where times at control points are decision variables, Carey [9] extended the single track line model to a double-track line model. Higgins et al. [18] provided a lower bound estimate of the remaining delay, in order to reduce the search space in the branch and bound procedure. Dessouky et al. [13] proposed adjacent propagation and feasibility propagation in the branch and bound procedure for a complex railway infrastructure. Zhou and Zhong [33] presented several methods to obtain lower bounds for a branch and bound procedure in order to solve the train scheduling problem in a single track line system. Mu and Dessouky [28] proposed several heuristic algorithms (generic algorithm, decomposition algorithm and parallel algorithm) and compared them with a benchmark given by a simple look-ahead greedy heuristic and a global neighborhood search heuristic. Li et al. [21] formulated a nonlinear mixed-integer programming model for a single-track railway corridor with the objective of minimizing the total delay. They proposed a heuristic method using a global conflict distribution prediction (CDP) to identify near-optimal travel strategies that are deadlock free.

In the second stream based on a time-space graph, Caprara et al. [6] introduced a time-space model for the train timetabling problem and derived a Lagrangian relaxation based method with a train fixing and removing heuristic to solve the Train Timetable Problem (TTP) for a single line railway system. Caprara et al. [8] extended the previous model by introducing several practical constraints like station capacity constraints. Cacchiani et al. [5] extended the time-space graph model to a general railway network. Talebian and Zou [31] studied strategic planning for passenger trains and freight trains on shared corridors. They proposed to sequentially optimize passenger trains’ delays and freight trains’ lost demand. The quadratic integer programming (QIP) formulation for passenger trains’ optimization is approximated through linearization. Zhou and Teng [35] formulated an integer linear programming for simultaneous planning of both the passenger train routing and timetabling problem. A Lagrangian relaxation decomposition framework together with a heuristic method is proposed.
Recently, another method of modeling called alternative graph, which can be traced back to Mascis and Pacciarelli [25], has been typically used for the train conflict detection and resolution (CDR) problem. CDR resolves train conflicts by modifying dwell times, train speeds and the schedule (Corman et al. [111]). The order of passing one track for two trains is modeled as two alternative arcs. Mazzarello and Ottaviani [26] extended the alternative graph model by considering constraints imposed by the real railway environment and embedded a simple model in a rolling time horizon scheme. D’Ariano et al. [12] presented a branch and bound procedure to solve the conflict resolution problem using an alternative graph model. Several heuristics were also proposed to speed up the computation. Novel tabu search neighborhood structures and search strategies were proposed in Corman et al. [111] to improve the solution quality and computation time.

Notice that in all models mentioned above, the railway track is discretized and represented as a set of blocks. In order to avoid collision, “on each block, there can only be one train at each time period” according to Brannlund [4]. Therefore headway between two consecutive trains will be maintained to be at least one block before the succeeding train enters a new block. This is a reasonable assumption for the traditional railway signal system since decisions are made at the control points, which are between two consecutive blocks. However, for a PTC system where dynamic headway may be adopted this may lead to inefficiencies. This is the gap this paper is trying to fill in.

3. Dynamic headway

Before we present our optimization model of the train scheduling and routing problem for dynamic headway, we first review the modelling of the train movement process for constant headway and then extend it for dynamic headway.

3.1. Constant headway model representation

In [22], the railway track is discretized into different segments. Segments are the smallest, indivisible units in this model. All points within one segment share one speed limit. Then several segments and/or junctions are grouped into one node to formulate a network $G = (V, E)$, where $V$ is the set of nodes or vertices and $E$ is the set of arcs. Notice that each arc works as linkage between two nodes, and that it may or may not correspond to a junction. An example network construction for a portion of the railway network is given as follows:

![Network Construction Example](image)

There are several requirements for this model:

1. There are no junctions within the segment. Junctions can only appear in the arcs or at the beginning or end of a segment within a node.
2. Each segment has the capacity of one, i.e. there can be at most one train within each segment at any time.
3. Arcs corresponding to junctions have certain speed limits while other arcs have no speed limits.
4. The length of a segment is no more than the maximum train length

Segments can be defined in different ways. However, if the segment is too long, the capacity of the network system will decrease since the capacity of each segment is set to be one (i.e. at most one train can occupy a segment at a time). Note in the above network definition a node may contain multiple segments since the segments may be much smaller than the length of a train.
Trains travelling through the railway network are modeled as movements through the nodes on the constructed graph. Since all the arcs have length of zero, the train’s travelling time can be divided into travelling time on the nodes along the taken path. On the constructed graph \( G = (V, E) \) a path \( P \) is defined as a sequence of nodes \( n_1, n_2, ..., n_m \), where \( n_1 \) is the origin and \( n_m \) is the destination. Let \( T(P) \) be the function denoting the minimum travelling time if the train takes path \( P \). Then

\[
T(P = n_1, n_2, ..., n_m) = \sum_{i=1}^{m} T(P = n_i)
\]

Therefore to obtain the minimum travelling time along a path, we need to compute the minimum travelling time within a single node. It is easy to compute \( T(n_i) \) if we don’t consider the trains’ dynamics. In other words, trains can reach any speed instantly since deceleration/acceleration rates are infinite. Then the travelling time through each segment is the ratio between the segment length and the speed limit. Therefore the travelling time within one node is the sum of the travelling times through the segments contained in this node. Suppose node \( n_i \) contains \( \rho_i \) segments, whose distances are \( l_1, l_2, ..., l_{\rho_i} \) and speed limits are \( \bar{v}_1, \bar{v}_2, ..., \bar{v}_{\rho_i} \). Then the simple way to estimate \( T(n_i) \) is

\[
T(n_i) = \sum_{j=1}^{\rho_i} l_j / \bar{v}_j
\]

However, the assumption of infinite deceleration/acceleration rates is too strong. Hence, the estimation of the train’s travel time without considering the trains’ dynamics leads to inaccurate results. Accordingly, the model cannot reflect the actual performance of the railway network.

To make a better estimation of a train’s travel time, [22] introduced trains’ dynamics to the model. Trains are assumed to have constant deceleration rates and maximal acceleration rates. Given these assumptions, they derive the following equations to compute the minimal travel time through a segment. Let the entering velocity, the exiting velocity, the segment distance and the speed limit be \( v_{\text{enter}}, v_{\text{exit}}, l \) and \( \bar{v} \) respectively and the acceleration (deceleration) rate be \( r_a (r_d) \). Then the minimal travel time \( t(v_{\text{enter}}, v_{\text{exit}}, l, \bar{v}) \) is:

1. If \( v_{\text{enter}} \leq \bar{v}, v_{\text{exit}}^2 - v_{\text{enter}}^2 = 2r_d l \)
   \[ t(v_{\text{enter}}, v_{\text{exit}}, l, \bar{v}) = \frac{v_{\text{exit}} - v_{\text{enter}}}{r_d} \]

2. If \( v_{\text{exit}} \leq v_{\text{enter}} \leq \bar{v}, v_{\text{exit}}^2 - v_{\text{enter}}^2 = 2r_d l \)
   \[ t(v_{\text{enter}}, v_{\text{exit}}, l, \bar{v}) = \frac{v_{\text{enter}} - v_{\text{exit}}}{r_d} \]

3. If \( v_{\text{exit}} \leq \bar{v}, v_{\text{enter}} \leq \bar{v}, v_{\text{exit}}^2 - v_{\text{enter}}^2 \leq 2r_d l, v_{\text{exit}}^2 - v_{\text{enter}}^2 \leq 2r_d l, \sqrt{\frac{r_a v_{\text{exit}}^2}{r_d} + \frac{r_d v_{\text{enter}}^2}{r_a} + \frac{2r_d l}{r_a}} \leq \bar{v} \)
   \[ t(v_{\text{enter}}, v_{\text{exit}}, l, \bar{v}) = \frac{v_{\text{enter}}}{r_a} - \frac{v_{\text{exit}}}{r_d} + \left( \frac{1}{r_a} + \frac{1}{r_d} \right) \sqrt{\frac{r_a v_{\text{exit}}^2}{r_d} + \frac{r_d v_{\text{enter}}^2}{r_a} + \frac{2r_d l}{r_a}} + \frac{1}{r_d} \]

4. If \( v_{\text{exit}} \leq \bar{v}, v_{\text{enter}} \leq \bar{v}, v_{\text{exit}}^2 - v_{\text{enter}}^2 \leq 2r_d l, v_{\text{exit}}^2 - v_{\text{enter}}^2 \leq 2r_d l, \sqrt{\frac{r_a v_{\text{exit}}^2}{r_d} + \frac{r_d v_{\text{enter}}^2}{r_a} + \frac{2r_d l}{r_a}} > \bar{v} \)
   \[ t(v_{\text{enter}}, v_{\text{exit}}, l, \bar{v}) = \frac{\bar{v} - v_{\text{enter}}}{r_a} + \frac{\bar{v} - v_{\text{exit}}}{r_d} + \frac{1}{r_a} \left( \frac{\bar{v}^2 - v_{\text{enter}}^2}{2r_a} - \frac{\bar{v}^2 - v_{\text{exit}}^2}{2r_d} \right) \]

5. Otherwise,
   \[ t(v_{\text{enter}}, v_{\text{exit}}, l, \bar{v}) = +\infty \]

In the first scenario, the train keeps accelerating from \( v_{\text{enter}} \) to \( v_{\text{exit}} \) while in the second scenario the train keeps decelerating from \( v_{\text{enter}} \) to \( v_{\text{exit}} \). In the third scenario, the train first accelerates from \( v_{\text{enter}} \) to a velocity, which is no more than the speed limit, and then decelerates to \( v_{\text{exit}} \). In the fourth scenario, the train accelerates from \( v_{\text{enter}} \) to the speed limit, and then travels at a constant speed, which is the speed limit, and then decelerates from the speed limit to \( v_{\text{exit}} \). Otherwise, the given \( v_{\text{enter}}, v_{\text{exit}}, l \) and \( \bar{v} \) are infeasible, which implies the travel time should be infinity.

After introducing the equations to obtain the travel time through a single segment, [22] proposed a velocity augmenting algorithm to compute the minimum travel time through several segments with different speed limits. Given the starting velocity, ending velocity, speed limits and segment lengths, the velocity augmenting algorithm can...
determine the velocity at the end of each segment, and then obtain the minimum travel time in $O(\rho^2)$, where $\rho$ is the number of segments.

In the model introduced above, a train’s movement is determined by its route and entering velocities at nodes along its route. Notice that for two consecutive nodes the exiting velocity for the preceding node is equal to the entering velocity for the succeeding node. It is difficult to make the optimal decision on the route and the entering velocities jointly. Thus, [22] separated these decisions by ensuring that a train can stop within each node it travels through. This can be achieved by the following two ways:

1. Make one node’s distance long enough
2. Make a train’s entering velocities low enough

Since the second way will slow down trains (if node distances are small), [22] constructed the network so that the distance of each node is long enough for trains to stop within each node. Then within each node a train’s movement decision is made to indicate the exiting velocity of the current node. Notice that if the train needs to stop within the current node in order to avoid collision or deadlock, the exiting velocity of the current node should be zero. Otherwise the exiting velocity can be determined through the velocity augmenting algorithm mentioned above.

3.2. Dynamic headway model representation

The model above can represent a train’s movement accurately. However the constraint that each node should be long enough for trains to stop within each node makes the network representation not very efficient since the capacity for each node is set to one. In other words, the succeeding train cannot enter the next node until the preceding train leaves. Therefore the headway between two consecutive trains is at least a node’s distance. Also notice that the headway is controlled by a node’s distance for all trains, which implies that this model can only provide “fixed” headway. We show a new reformulation for “dynamic” headway, i.e. the headway between two consecutive trains is dependent on the two trains’ types and velocities.

Similar to the previous model, the railway track is divided into small segments. Then a network $G = (V, E)$ is constructed where each node in $G$ corresponds to one segment and each arc in $G$ corresponds to one connection between two segments. Notice that one arc may or may not correspond to a junction in the railway network. Then a train’s movement through the railway network can be modelled as movement through the constructed network $G$. Again we set the capacity of each node to be one, i.e. there can be at most one train within each node at any time. Then the headway is modeled as all the available nodes between two consecutive trains.

Different from the previous model, we do not constrain each node’s length. The smaller each node’s length is, the finer our discretization and approximation for the headway is. On the other hand, the number of nodes in the constructed network will increase as the nodes’ length decrease. Also we do not enforce a train must stop within its current node. Otherwise the entering velocities must be small enough and therefore trains may never reach the speed limit. Therefore a certain number of nodes ahead of each train must be assigned to it before the train can enter the current node. We call these assigned nodes to a train as its dynamic headway. Dynamic headway should be long enough in order for the corresponding train to come to a full stop without collision.

The dynamic headway is categorized into two types. In the first scenario, there is no preceding train within the focal train’s braking distance. Then the dynamic headway is no shorter than the braking distance. Let the focal train’s velocity be $v$ and its maximal deceleration rate be $r_{d1}$. Then the dynamic headway distance $H_D$ is:

$$H_D \geq \frac{v^2}{2r_{d1}}$$

In the second scenario, the succeeding train’s dynamic headway works as buffer between the two consecutive trains. Let the preceding train’s velocity be $\mu$, its maximal deceleration rate be $r_{d2}$ and the response time for the preceding train be $\Delta t_1$. Then the dynamic headway distance $H_D$ should be long enough to avoid collision, i.e.

$$H_D + \frac{\mu^2}{2r_{d2}} \geq \frac{v^2}{2r_{d1}} + \Delta t_1 \ast v$$

Thus,

$$H_D \geq \frac{v^2}{2r_{d1}} + \Delta t_1 \ast v - \frac{\mu^2}{2r_{d2}}$$
3.3. Problem formulation for optimization model

To effectively model dynamic headway, each node length has to be small and hence each train occupies several nodes at the same time. Also each train can occupy some nodes ahead of it as a buffer between it and its preceding train. These nodes are defined to be “headway nodes”. Since the headway nodes need to be determined by the trains’ velocities and deceleration/acceleration rates, the number and the length of headway nodes vary as the trains are traveling through the network. We name it as dynamic headway control. The train scheduling problem for dynamic headway control seeks a path for each train, controls each train’s velocity along the path and assigns headway nodes in an efficient way. We now present a mathematical formulation for this problem.

For the optimization model formulation, we make three simplifying assumptions.

1. All the nodes in the network have the same length.
2. The length of each train is a multiple of a node length.
3. All headway nodes work as braking distance.

Note that when we present the heuristics in section 4, we relax these assumptions.

Index Notation

\[ i, j \text{ Node index} \]
\[ k \text{ Train index} \]

Parameters

\[ G = (V, E) \text{ a directed network representing the railway system, } V \text{ is the node set and } E \text{ is the arc set} \]
\[ K \text{ the train set} \]
\[ s_k \text{ the origin node for train } k \]
\[ d_k \text{ the destination node for train } k \]
\[ N_k \text{ the ratio between the length of train } k \text{ and the node length} \]
\[ \bar{v}_{ik} \text{ the speed limit on node } i \text{ for train } k \]
\[ l_0 \text{ the length of each node} \]
\[ \Gamma_{ik}(v_{\text{enter}}, v_{\text{exit}}, v') \text{ the traveling time of train } k \text{ through node } i \text{ as a function of the arrival velocity } v_{\text{enter}}, \text{ departure velocity } v_{\text{exit}} \text{ and the speed limit } v' \]
\[ \text{BR}_k(v) \text{ the braking distance for train } k \text{ traveling at velocity } v \]
\[ \delta_{ik} \begin{cases} -1 & \text{if } i = s_k \\ 1 & \text{if } i = d_k \\ 0 & \text{Otherwise} \end{cases} \]
\[ M \text{ a sufficiently large number} \]
\[ W_L \text{ the set of paths containing } L \text{ nodes in } G \]
\[ L_{\text{max}} \text{ the maximum number of nodes one train can occupy at any time} \]

1) \( \Gamma_{ik}(v_{\text{enter}}, v_{\text{exit}}, v') \) and \( \text{BR}_k(v) \) are general functions for the traveling time and the braking distance. We can treat the functions introduced in section 3.1 and 3.2 as special cases. Formally, under the assumption of constant deceleration/acceleration rates,

\[ \Gamma_{ik}(v_{\text{enter}}, v_{\text{exit}}, v') = t(v_{\text{enter}}, v_{\text{exit}}, l_0, v') \]

and

\[ \text{BR}_k(v) = \frac{v^2}{2r_{kd}} \]

where \( r_{kd} \) is the deceleration rate for train \( k \).

2) \( L_{\text{max}} \) is the maximum number of occupied nodes, including the headway nodes. One reasonable way to set \( L_{\text{max}} \) is

\[ L_{\text{max}} = 2 \max_{k \in K} N_k \]

which assumes that the total length of the headway nodes is no longer than the train length.

Variables

\[ f_{ijk} = \begin{cases} 1 & \text{if train } k \text{ travels from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases} \]
\[ p_{i_1, i_2, \ldots, i_L, k} = \begin{cases} 1 & \text{if train } k \text{ takes the path } (i_1, i_2, \ldots, i_L) \in W_L \\ 0 & \text{otherwise} \end{cases} \]
\[ y_{ijk} = \begin{cases} 1 & \text{if the next headway node is } j \text{ when the head of the train } k \text{ enters node } i \\ 0 & \text{otherwise} \end{cases} \]
The arrival velocity when the head of train $k$ enters node $i$

$v^A_{ik}$

The maximal velocity train $k$ can travel when the head of train $k$ is within node $i$

$v^H_{ik}$

the arrival time when the head of train $k$ enters node $i$

$t^A_{ik}$

the departure time when the tail of train $k$ leaves node $i$

$t^D_{ik}$

the arrival time when the train occupies node $i$ for the first time

$x_{k,i,k} = \begin{cases} 
1 & \text{if train } k \text{ occupies node } i \text{ before train } k_2 \\
0 & \text{otherwise}
\end{cases}

Note that $v^A_{ik}$ is different from $v^H_{ik}$ since train $k$ must travel under all the speed limits for its physically seized nodes.

The path variables $P_{i_1, i_2, ..., i_k}$ and $v^A_{ik}$ are needed to determine $v^H_{ik}$.

**Formulation**

\[
\text{Minimize} \quad \sum_{k \in K} (t^A_{d_k} - t^D_{d_k})
\]

Subject to

\[
\sum_{j \in (i,j) \in E} f_{ijk} + \sum_{j \in (j,j) \in E} f_{ijk} = \delta_{ik} \quad \forall k \in K, \forall i \in V
\]

\[
P_{i,k} = \sum_{f \in (i,j) \in E} f_{ijk} \quad \forall k \in K, \forall i \in V
\]

\[
P_{i_1,i_2,...,i_k} = \prod_{l=1}^{L-1} f_{i_l,i_{l+1}} \quad \forall k \in K, \forall (i_1,i_2,...,i_L) \in W_L, \forall L = 2,...,L_{max}
\]

\[
v^A_{x_k} = 0 \quad \forall k \in K
\]

\[
v^A_{d_k} = 0 \quad \forall k \in K
\]

\[
\sum_{j \in (i,j) \in E} f_{ijk} v^A_{j} - t^H_{ik} = \Gamma_{ik} \left( v^A_{ik}, \sum_{j \in (j,j) \in E} f_{ijk} v^A_{j}, 0 \right) \quad \forall k \in K, \forall i \in V
\]

\[
P_{i_1,i_2,...,i_k} v^A_{i_1,k} \geq v^A_{i_k} \quad \forall k \in K, \forall i \in V, \forall L = 1,2,...,N_k + 1
\]

\[
P_{i_1,i_2,...,i_k, k} v^A_{i_1,k} = t^A_{ik} \quad \forall k \in K, \forall i \in V
\]

\[
\sum_{j \in (i,j) \in E} y_{ijkl} t^H_{ik} \leq t^A_{ik} \quad \forall k \in K, \forall i \in V
\]

\[
\sum_{j \in (i,j) \in E} f_{ijk} t^H_{ik} \geq t^H_{ik} \quad \forall k \in K, \forall i \in V
\]

\[
(1 - P_{i,k}) M \geq 1 - \sum_{j \in (i,j) \in E} y_{ijkl} \quad \forall k \in K, \forall i \in V
\]

\[
\sum_{j \in (i,j) \in E} y_{ijkl} \leq 1 \quad \forall k \in K, \forall i \in V
\]

\[
(1 - y_{ijkl}) M \geq 2 - P_{i,k} - P_{i,j} \quad \forall k \in K, \forall i \in V
\]

\[
\sum_{j \in (i,j) \in E} \sum_{L=1}^{L_{max}} L P_{i_1,i_2,...,i_L} y_{ijkl} l_0 \geq \beta R_k(v^A_{i_1,k}) \quad \forall k \in K, \forall i \in V
\]

\[
(1 - x_{k_1,k_2}) M + t^H_{ik_1} \geq t^H_{ik_2} \quad \forall k_1,k_2 \in K, \forall i \in V
\]

\[
(2 - P_{i,k_1} - P_{i,k_2}) M \geq 1 - x_{k_1,k_2} + x_{k_1,k_i} \quad \forall k_1,k_2 \in K, \forall i \in V
\]

\[
x_{k_1,k_2} \leq 1 \quad \forall k_1,k_2 \in K, \forall i \in V
\]

\[
x_{k_1,k_2} \leq 1 \quad \forall k_1,k_2 \in K, \forall i \in V
\]

\[
x_{k_1,k_2} \in \{0,1\} \quad \forall k_1,k_2 \in K, \forall i \in V
\]

\[
i_{j_{ik}} \geq 0 \quad \forall k \in K, \forall i \in V
\]

\[
P_{i_1,i_2,...,i_k} \in \{0,1\} \quad \forall k \in K, \forall (i_1,i_2,...,i_L) \in W_L, \forall L = 1,...,L_{max}
\]

\[
v^A_{ik},v^A_{d_k},t^A_{ik},t^D_{ik} \in \mathbb{R}^+ \quad \forall k \in K, \forall i \in V
\]
The objective is to minimize the total travel time for all the trains. Constraint (1) ensures the flow conservation for each train. Constraints (2) and (3) give the relationship between \( f_{ijk} \) and \( P_{ij_l...i_{jk}} \). Note \( P_{ij_l...i_{jk}} \) is introduced to cope with the fact that one train can seize multiple nodes at the same time. \( L = 1 \), i.e. \( P_{jk} \) indicates whether train \( k \) travels through node \( i \), which is used in (2), (11), (13) and (16). When \( L \geq 2 \), \( P_{ij_l...i_{jk}} \) indicates whether train \( k \) takes the path \( i_l i_{2}...i_L \), which is used in (3), (7), and (14). Constraints (4) and (5) specify the initial velocity at the origin node and the final velocity at the destination node for each train must be zero.

Constraints (6) and (7) deal with the travel time across a node. In constraint (6), \( \sum_{f \in E} f_{ijk} t_{jk}^f \) is the time when the train’s head exits from node \( i \) (i.e. the train’s head enters the next node) and \( \sum_{f \in E} f_{ijk} v_{jk}^f \) is the exiting velocity for the train’s head. Thus, constraint (6) ensures the travel time across node \( i \) is based on the given function \( f_{ijk} \), where \( v_{jk}^f \) is the maximal velocity without violating the speed limits when the train’s head is within node \( i \). Constraint (7) sets \( v_{jk}^f \) to be the minimum of all the speed limits for the nodes the train physically seizes, where \( \sum_{f \in E} f_{ijk} v_{jk}^f \) is the speed limit for the \( L^{th} \) node away from node \( i \) (node \( i \)). For example, when \( L = N_k + 1 \), node \( i_{N_k+1} \) is node \( i \) as shown in Figure 2. Then constraint (7) becomes \( \bar{v}_{i_{N_k+1}} \geq v_{jk}^f \), i.e. when the train’s head is within node \( i_{N_k+1} \), it must travel under \( \bar{v}_{i_{N_k+1}} \), the speed limit of node \( i_{N_k+1} \).

Figure 2 Speed Limit, Departure and Arrival Time

Constraints (8) specifies the relationship between the train tail’s departure time and the train head’s arrival time. Recall that we made the assumption that the train length is a multiple of the node length \( l_0 \). So the arrival time to the next node \( i_{N_k+1} \) is equal to the departure time from the last node \( i_0 \) the train physically seizes, as shown in Figure 2. Notice that the length of path \( (i_0, i_1, ..., i_{N_k+1}) \) is \( N_k + 2 \).

Constraints (9)-(14) deal with the choice of the headway nodes. \( \sum_{j \in V} y_{ijk} t_{jk}^{HW} \) is the time when a train occupies a new node as its headway node for the first time. So constraint (9) ensures that a train can only occupy a new headway node when its head enters into a new node (node \( i \)). Given the path, a train needs to occupy the headway nodes along the path in its sequence, which is satisfied by (10). Constraints (11) and (12) reveal the relationship between \( P_{ijk} \) and \( y_{ijk} \). If train \( k \) travels through node \( i \) (\( P_{ijk} = 1 \)), then a headway decision should be made at node \( i \) (\( \sum_{j \in V} y_{ijk} = 1 \)). Conversely, if a headway decision is made at node \( i \) (\( y_{ijk} = 1 \)), both \( i \) and \( j \) should be visited by train \( k \) (\( P_{ijk} + P_{ijk} = 2 \)), which is validated in (13). Constraint (14) ensures that the headway is long enough for trains to come to a full stop, where \( \sum_{j \in V} \sum_{i=0}^{L_{max}} \sum_{f \in E} f_{i_l...i_{jk}} \) is the total length of all the headway nodes, as indicated in Figure 3.

Figure 3 Braking Distance.

Constraints (15)-(17) deal with the scheduling problem when multiple trains travel through the same network. Constraint (15) ensures that a node can only be occupied as a headway node only if no other trains occupy it.
Constraints (16) and (17) specify that if two trains try to visit one node \((P_{ik1} + P_{ik2} = 2)\), they should visit this node in sequence \((x_{k1,k,i} + x_{k2,k,i} = 1)\).

Constraints (18) and (19) state that the three events: occupied as a headway node \((t_{ik}^{AH})\), train arrival \((t_{ik}^A)\) and train departure \((t_{ik}^D)\) must occur in sequence. Finally, constraints (20)-(23) specify the domain of the decision variables.

From the formulation above, the train scheduling problem for dynamic headway control is a Mixed Integer Programming problem with nonlinear constraints. It makes this problem difficult to solve, especially when the network is large. Instead of solving this problem exactly and directly using the formulation, we (1) consider the problem of scheduling each train iteratively, i.e., route a new train from its origin to its destination given the other trains’ schedules and (2) decompose the entire problem into three subproblems based on the decision variables which are classified into three types: headway decision \((y_{ijk})\), velocity decision \((v_{ik}^f)\) and routing and scheduling decision \((f_{ijk}, P_{i_{1,i_{2-i},k}}, t_{ik}^A)\).

In section 4, we present a heuristic approach to deal with the headway decision and velocity decision. For the routing and scheduling decision, we use the same greedy algorithm as in [22].

4. Heuristics for dynamic headway

In this section, we present our solution procedure for the velocity and headway decisions for dynamic headway control. In general, the procedure provides a dynamic control scheme for headway control. Decisions are made at certain points (at the beginning of each segment). We call them decision points. When a train reaches a decision point, the solution procedure is applied to obtain the velocity at the next decision point and the corresponding travel time.

Notice that the following decisions need to be made to determine the train’s movement in the dynamic headway model:

1. Routing decision, i.e. the choice of the next headway node if there is more than one candidate
2. Headway decision, i.e. the number of new headway nodes a train needs to occupy
3. Velocity decision, i.e. the velocity at the next decision point

For simplicity, in this section we focus on the last two problems. For the first decision, we assume a greedy routing algorithm, that is, we route to the next headway node which has the highest speed limit. We first determine the number of headway nodes we need and then calculate the velocity at the next decision point based on the new headway information.

4.1. Headway decision

We describe a simulation approach for modelling and solving the headway decision. Based on the headway, each train will be assigned several nodes ahead of it in the simulation model. When the train moves towards the next headway node, the simulation model will determine whether new headway nodes are needed and assign them to the train if necessary. Moreover, the exiting velocity of the next headway node needs to be calculated. In summary the simulation works as follows:

1. When a train enters into the first node of its schedule, it is assigned several nodes to serve as the headway between it and its preceding train.
2. When a train enters into the first headway node (nearest node to the train in the headway nodes), the routing algorithm will be called to determine the headway nodes if needed. Also the exiting velocity of the first headway node will be calculated. Then the train will be routed to the end of the first headway node according to the exiting velocity.
3. When the last headway node (farthest node to the train in the headway nodes) reaches the end node of the schedule. The train is routed to a full stop at the end node according to its minimal travel time.

Notice that whenever new headway nodes are added, they must not cause deadlock. (According to [22], a typical deadlock situation occurs when two trains running in the opposite direction are routed to nodes representing the same single-track line simultaneously.) Also when there are many choices for the new headway nodes, which are all deadlock free, good decisions of which headway nodes to select need to be made. A deadlock free routing heuristic together with appropriate choice of exiting velocities will be discussed in the next section.

We now show an example to illustrate the difference between the headway control for constant headway and the new dynamic headway control. In the fixed headway simulation modelling approach, a train needs to decelerate to a full stop if the train finds the next node is held by other trains. The following graph demonstrates how dynamic headway works against the fixed headway model. In Figure 4 when train 1 enters Node A, it prepares to decelerate to avoid entering Node B which is occupied by train 2. Figure 5 shows the logic behind the dynamic headway schema for the same track configuration. Train 1 pre-occupies Node 1 and Node 2 (in other words, train 1 seizes Node 1 and
Node 2 even before it physically enters into them). When train 1 enters Node 1, the routing algorithm is called and Node 3 is added to train 1’s pre-occupied nodes. Therefore train 1 does not need to decelerate when traveling within Node 1.

Figure 4 Fixed Headway Model.

Figure 5 Dynamic Headway Model.

The headway decision problem is used to determine the number of headway nodes to seize given the current velocity at a decision point. Formally, when a train reaches the beginning of node \( n_0 \), its velocity is \( v \) and its currently seized headway nodes are \( \{n_0, n_1, ..., n_{Q_0} - 1\} \), where \( Q_0 \) is the number of headway nodes. We name node \( n_0 \) as the current node. If the next headway node candidate is \( n_{Q_0} \) and suppose that the speed limit for node \( n_i \) is \( \bar{v}_i \) (\( i = 0, 1, ..., Q_0 \)), the problem is to decide whether to add \( n_{Q_0} \) as the new headway node so that the train can travel as fast as possible. Actually after adding \( n_{Q_0} \) to the headway nodes, we can still ask whether to add \( n_{Q_0} + 1 \) as the next headway node. It turns out that we can sequentially add new headway nodes until the headway is “long enough”. Therefore the headway decision problem is to determine how many new headway nodes are needed.

With “long enough” headway, in a scenario where new headway nodes are not needed, a train can reach the speed limit at some certain node and then can still come to a full stop within the headway nodes. However if the train can only reach the speed limit at the current node, it might travel under the speed limit when existing the current node. Thus, the train can travel faster within the current node given more headway nodes. Therefore we define the headway to be “long enough” if the train can reach the speed limit outside the current node.

A simple example is given in Figure 6, which has the same network configuration as in Figure 5, to illustrate the concept of “long enough” headway, where we plot the train’s travel speed at different track segment locations. Suppose all nodes share the same speed limit. A train just reaches a decision point (the beginning of node 1) and occupies node 1 and node 2 as the current headway nodes. If node 3 is not available and there are no trains traveling ahead of it in
the same direction, the train will decelerate as shown in line 1 to ensure that the train can stop before it reaches node 3. If node 3 is available, we can seize node 3 and the train will move as in line 2. Even though the speed limit is reached within node 1, at point P which is at the end of node 1, the train must travel under the speed limit in order to ensure that it can stop at the end of node 3. If the train seizes another headway node (node 4), it will move along as in line 3 and can travel at the maximum speed for a longer stretch of time. Therefore nodes 1, 2, 3 and 4 together are “long enough” since the train can get to the speed limit at node 2, which is not the current node.

As comparison to the fixed headway control rules, if we modeled nodes 1 to node 4 as a single node in the simulation model, we would be able to ensure that this train would travel at its maximum speed for a longer stretch of time, but this would come at the expense of the fact that this track segment would be held by this train longer than necessary, thus needlessly reducing system capacity.

4.2. Velocity decision

Recall that in section 3.2 we introduce two different types of headway: the first type of headway works as braking distance and the second type of headway works as buffer distance between two consecutive trains. We show two methods of obtaining the velocity at the end of the current node accordingly. We first consider the scenario where the headway works as the braking distance, i.e. there are no preceding trains.

Since we make the decision dynamically, only the velocity at the end of the current node needs to be determined, and then the train travels through the current node. Given the current node \( n_0 \) and the current velocity \( v_0 \), to minimize the travel time within the current node, we want to maximize the velocity at the end of the current node since the function \( t(v_{\text{enter}}, v_{\text{exit}}, l, \bar{v}) \) given in section 3.1 decreases as \( v_{\text{exit}} \) increases. Let \( v_1^* \) be the optimal exiting speed of the train’s head from node \( n_0 \), i.e. the optimal entering speed to node \( n_1 \). Suppose after making the headway decision, the headway nodes are \( n_1, n_2, ..., n_Q \), so the number of seized headway nodes is \( Q \). Let the velocity at the beginning of node \( i \) be \( v_i \) \((i = 1, ..., Q)\) and let the velocity at the end of node \( n_Q \) be \( v_{Q+1} \). Consider a sequence of problems indexed by \( \beta \) \((\beta = 1, ..., Q)\):

\[
\begin{align*}
\text{max} & \quad v_\beta \\
\text{s.t.} & \quad t(v_i, v_{i+1}, l_i, \bar{v}) < +\infty \quad \forall i = \beta, ..., Q \\
& \quad v_i \leq \bar{v}_i \quad \forall i = \beta, ..., Q \\
& \quad v_{Q+1} = 0
\end{align*}
\]

The \( \beta^{th} \) problem solves the maximal velocity at the beginning of node \( n_\beta \) so that the train can stop at the end of node \( n_Q \). The first constraint makes it feasible for a train to travel through node \( i \) at a velocity of \( v_i \) at the beginning of the node to a velocity of \( v_{i+1} \) at the end of the node \((i = \beta, ..., Q)\).

Let \( \bar{v}_\beta \) be the optimal value for the \( \beta^{th} \) problem \((\beta = 1, ..., Q)\). Intuitively, the above sequence of problems recursively solve for \( \bar{v}_1 \) when the train is at the beginning of node \( n_1 \). The \( \beta^{th} \) problem can take advantage of the optimal value obtained by the \((\beta + 1)^{th} \) problem instead of solving the \( \beta^{th} \) problem explicitly. From \( t(v_i, v_{i+1}, s_i, l_i) < +\infty \), we know \( v_i \leq \sqrt{v_{i+1}^2 + 2 r_a l_i} \). So we have \( \forall \beta = 1, ..., Q \)

\[
\bar{v}_\beta \leq \sqrt{v_{\beta+1}^2 + 2 r_a l_\beta} \leq \sqrt{\bar{v}_{\beta+1}^2 + 2 r_a l_\beta}
\]

where \( v_{\beta+1} \) is the velocity at the beginning of \( n_{\beta+1} \) given by the optimal solution for the \( \beta^{th} \) problem. The second inequality holds since \( v_{\beta+1} \) also gives a feasible solution for the \((\beta + 1)^{th} \) problem, which implies \( v_{\beta+1} \leq \bar{v}_{\beta+1} \)

Therefore

\[
\bar{v}_\beta = \min \left\{ \bar{v}_\beta, \sqrt{\bar{v}_{\beta+1}^2 + 2 r_a l_\beta} \right\} \quad \forall \beta = 1, ..., Q
\]

So we can obtain \( \bar{v}_1 \) backwards by first obtaining \( \bar{v}_Q \), and then applying the above equation recursively until we arrive at \( \beta = 1 \).

Finally, we also need to take the travel procedure within node \( n_1 \) into consideration. To satisfy \( t(v_0, v_1, l_1, \bar{v}_1) < +\infty \), we need to take the minimal of \( \sqrt{v_0^2 + 2 r_a l_1} \) and \( \bar{v}_1 \). Therefore the maximal feasible velocity \( v_1^* \) at the end of the next node should be

\[
v_1^* = \min \{ \bar{v}_1, \sqrt{v_0^2 + 2 r_a l_1} \}
\]

Then we consider the other scenario where the headway works as buffer distance between two successive trains. Again let the current headway nodes be \( n_1, n_2, ..., n_Q \) and node \( n_{Q+1} \) is occupied by the preceding train which travels in the same direction. Let the current velocity of the preceding train be \( \mu \). Then we can obtain the potential maximal velocity \( \bar{v}_1 \) at the beginning of node \( n_1 \) as follows:
\[ \sum_{i=1}^{Q} l_i = \tilde{v}_1^2 / (2r_{d1}) + \Delta t_1 \ast \tilde{v}_1 - \mu^2 / (2r_{d2}) \]

where \( \Delta t_1 \) is the response time for the succeeding train, \( r_{d1} \) and \( r_{d2} \) are the deceleration rates for the succeeding and preceding trains respectively.

However, \( \tilde{v}_1 \) obtained by the equation above may not be accurate. When the succeeding train travels \( \sum_{i=1}^{Q} l_i + \mu^2 / (2r_{d2}) \), it may stop within some node which is occupied by the preceding train. To obtain the maximal velocity \( v_1^* \), let

\[ \sum_{i=Q+1}^{R} l_i \leq \frac{\mu^2}{2r_{d2}} \leq \sum_{i=Q+1}^{R+1} l_i \]

where nodes \( n_{Q+1}, \ldots, n_R \) are occupied by the preceding train (some of them are physically occupied and others are headway nodes). Notice that \( R \) can be uniquely determined by the above inequalities, and therefore is dependent on \( \mu \). So the maximal velocity \( v_1^* \) at the beginning of node \( n_1 \) satisfies

\[ \sum_{i=1}^{R} l_i = \frac{v_1^2}{(2r_{d1})} + \Delta t_1 \ast v_1^* \]

Thus,

\[ v_1^* = \frac{r_{d1} \ast \sqrt{\Delta t_1^2 + \frac{2}{r_{d1}} \ast \sum_{i=1}^{R} l_i}}{2} \]

In summary, whenever a train enters into a node, two decisions will be made: how many new headway nodes are needed (headway decision) and the exiting speed for the current node (velocity decision). The train will seize these new headway nodes. And given the speed limit for the current node and the train’s deceleration and acceleration rates, we can calculate the minimal traveling time within the current node. The detailed steps are:

1. A train enters into a node, which is a previously seized headway node.
2. Is a new headway node available? If yes, go to 3. If no, go to 5.
3. Can the train reach the speed limit outside the current node given the current headway nodes? If yes, go to 6. If no, go to 4.
4. Seize one new headway node. Go to 2.
5. Is the new headway node seized by a train traveling in the same direction? If yes, go to 7. If no, go to 6.
6. Compute the exiting speed which can make the train stop within the current headway, go to 8.
7. Compute the exiting speed which can avoid collision with the preceding train.
8. Route the train to the end of its current node, where the train reaches the exiting speed.

5. Numerical experiments

To evaluate the proposed dynamic headway model and solution procedure, we present simulation experiments conducted on an actual railway network using the Arena simulation software. The chosen railway network is from Downtown Los Angeles to Pomona. Figure 7 illustrates the mileage between the stations. At the intermediate station (EL Monte), there is a crossing for trains traveling in the north/south direction. Notice that this figure only provides mileage information and does not show the actual railway trackage configuration. The railway trackage configuration in this area consists of single-, double- and triple- tracks.

Figure 7 Mileage Information.

Also two types of trains (freight train and passenger train) are tested on this area. The detailed information about these two types of trains can be found in Table 1.
Table 1 Characteristics of the Passenger and Freight Trains

<table>
<thead>
<tr>
<th></th>
<th>Length (feet)</th>
<th>Max speed (feet/min)</th>
<th>Acceleration rate (feet/min**2)</th>
<th>Deceleration rate (feet/min**2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freight Train</td>
<td>6000</td>
<td>6160</td>
<td>1584</td>
<td>1584</td>
</tr>
<tr>
<td>Passenger Train</td>
<td>1000</td>
<td>6952</td>
<td>2112</td>
<td>2112</td>
</tr>
</tbody>
</table>

We compared our dynamic headway modeling approach with the constant headway approach. The average node length in the constant headway model is 1.63 miles while in the dynamic headway model we tested the scenarios with average lengths of 0.83 and 0.55 miles, respectively.

5.1. One direction dispatching

We first tested the scenario with all trains travelling in the same direction (westbound from Pomona to Downtown Los Angeles) and with no crossing trains at the El Monte station. The simulation results are shown in Table 2. In the experiments, the number of trains is evenly divided between passenger and freight trains. For example, in the row with number of trains per day is 10 in Table 2, the arrival rate in the simulation is 5 per day for freight trains and 5 per day for passenger trains. The arrival process of both freight trains and passenger trains were assumed to follow a Poisson Process.

Table 2 Average Delay (Measured in Minutes) for the Different Approaches

<table>
<thead>
<tr>
<th>Number of trains per day</th>
<th>Constant headway method with average node length of 1.63 miles</th>
<th>Dynamic headway method with average node length of 0.83 miles</th>
<th>Dynamic headway method with average node length of 0.55 of miles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.40</td>
<td>0.44</td>
<td>0.42</td>
</tr>
<tr>
<td>20</td>
<td>0.86</td>
<td>0.67</td>
<td>0.69</td>
</tr>
<tr>
<td>40</td>
<td>1.94</td>
<td>1.76</td>
<td>1.59</td>
</tr>
<tr>
<td>60</td>
<td>3.07</td>
<td>3.04</td>
<td>2.64</td>
</tr>
<tr>
<td>80</td>
<td>4.55</td>
<td>4.73</td>
<td>3.83</td>
</tr>
<tr>
<td>100</td>
<td>6.45</td>
<td>7.20</td>
<td>5.33</td>
</tr>
<tr>
<td>150</td>
<td>15.28</td>
<td>23.10</td>
<td>11.86</td>
</tr>
<tr>
<td>170</td>
<td>28.08</td>
<td>88.36</td>
<td>16.15</td>
</tr>
</tbody>
</table>

The delay is measured as the difference between the actual travel time and the free travel time (i.e., the traveling time of the train when there are no other trains in the network). Since in the above example, all the trains are travelling in the same direction there are only two types of delay. The first type is at the beginning of the rail network where the first node (in the constant headway model) or set of nodes representing the headway (in the dynamic approach) is already occupied by a train so the new train cannot seize the track. The second type of delay is when a fast train (passenger) catches up with a slow train (freight).

From the above results, we can conclude:
1. In all cases, the average delay increases nonlinearly with increasing train arrivals. However, using a constant headway approach, the rail network is fully saturated when there are around 200 trains so we only range the number of trains up to 170. And if a dynamic headway approach is used the rail network capacity will be around 250 trains per day.
2. The dynamic headway approach with a smaller node size (0.55 miles) significantly outperforms the constant headway method when there is congestion in the network (the number of trains is large).
3. The dynamic headway approach provides lower average delay if we construct a network with a smaller node size. Since the railway system is represented as a node-arc discretized network, a smaller discretization of the network more closely represents the actual continuous process of train movement.
4. Sometimes the dynamic headway approach with a larger node size performs slightly worse than the constant headway method because in the dynamic headway approach we greedily seize headway nodes until a train reaches its maximal speed. If each node is relatively large, the trains may seize more railway track resources than necessary to ensure a safe headway when seizing the last headway node.
5.2. **Bi-direction dispatching**

We then tested the scenarios with trains traveling in both directions (eastbound from Downtown to Pomona and westbound from Pomona to Downtown). The results are shown in Table 3. In this case, the average train count per day is equally divided by each direction and type. Thus, for the row with 10 trains per day, the average number of freight trains travelling from Downtown Los Angeles to Pomona is 2.5. With trains travelling in both directions there are more chances for delay due to conflict with trains moving in opposing direction.

Table 3 Average Delay (Measured in Minutes) for Bi-Direction Dispatching

<table>
<thead>
<tr>
<th>Number of trains per day</th>
<th>Constant headway method with average node length of 1.63 miles</th>
<th>Dynamic headway method with average node length of 0.83 miles</th>
<th>Dynamic headway method with average node length of 0.55 miles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.40</td>
<td>1.15</td>
<td>1.21</td>
</tr>
<tr>
<td>20</td>
<td>2.87</td>
<td>2.78</td>
<td>2.74</td>
</tr>
<tr>
<td>40</td>
<td>5.49</td>
<td>5.29</td>
<td>5.27</td>
</tr>
<tr>
<td>60</td>
<td>9.42</td>
<td>8.91</td>
<td>8.85</td>
</tr>
<tr>
<td>80</td>
<td>13.29</td>
<td>12.80</td>
<td>12.69</td>
</tr>
<tr>
<td>100</td>
<td>20.68</td>
<td>17.37</td>
<td>16.90</td>
</tr>
<tr>
<td>150</td>
<td>55.44</td>
<td>43.72</td>
<td>32.02</td>
</tr>
</tbody>
</table>

From the above results, we can conclude:
1. Compared to the results when all the trains are traveling in the same direction, the average delay for trains traveling in both directions becomes larger no matter which method we use because there are more chances for conflict and congestion.
2. In general, the dynamic headway approach, even with a larger node size, performs much better than the constant headway method. The dynamic headway method has a smaller node size than the constant headway method and it can assign the smaller resources more efficiently to trains.

5.3. **Bi-Direction Dispatching with Crossing**

Finally we tested a scenario with a crossing at EL Monte. This time we only compare the conventional method and the new method with an average node length of 0.55 miles. The results are shown in Table 4. Numbers in the horizontal line denote the number of trains traveling northbound and southbound crossing EL Monte while numbers in the vertical line denote the number of trains traveling eastbound and westbound between Downtown and Pomona.

Table 4 Average Delay for Eastbound and Westbound Trains in Constant Headway Approach

<table>
<thead>
<tr>
<th>Number of trains per day</th>
<th>40</th>
<th>80</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3.12</td>
<td>4.81</td>
<td>7.60</td>
</tr>
<tr>
<td>20</td>
<td>4.31</td>
<td>6.73</td>
<td>10.16</td>
</tr>
<tr>
<td>40</td>
<td>7.77</td>
<td>10.83</td>
<td>14.94</td>
</tr>
<tr>
<td>60</td>
<td>12.10</td>
<td>16.71</td>
<td>22.30</td>
</tr>
<tr>
<td>80</td>
<td>18.05</td>
<td>24.47</td>
<td>34.25</td>
</tr>
</tbody>
</table>
With a crossing at EL Monte, the network capacity drops to less than 150 for the constant headway method. So we range the number of trains from 10 to 100. In contrast, the network capacity for the dynamic headway method is around 150 per day. Compared to the results when there are no crossing trains at El Monte, the average delay of eastbound and westbound trains is much larger due to the interference of southbound and northbound trains. Table 5 represents the ratios between the eastbound and westbound trains’ average delay for the dynamic headway method and the eastbound and westbound trains’ average delay for the constant headway method.

Table 5 Average Delay Ratio for Dynamic Headway Approach and Constant Headway Approach

<table>
<thead>
<tr>
<th></th>
<th>40</th>
<th>80</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.02</td>
<td>1.11</td>
<td>1.24</td>
</tr>
<tr>
<td>20</td>
<td>1.05</td>
<td>1.14</td>
<td>1.14</td>
</tr>
<tr>
<td>40</td>
<td>0.94</td>
<td>1.02</td>
<td>1.10</td>
</tr>
<tr>
<td>60</td>
<td>0.93</td>
<td>0.95</td>
<td>1.00</td>
</tr>
<tr>
<td>80</td>
<td>0.87</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>100</td>
<td>0.78</td>
<td>0.74</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Notice that the measure here is the average delay for the trains traveling in the eastbound and westbound directions. It can be seen sometimes when the number of trains traveling eastbound and westbound is smaller than the number of trains traveling northbound and southbound, the dynamic headway method is slightly worse than the constant headway method (the ratio is greater than 1). Since in this scenario the number of northbound and southbound trains is greater than the number of eastbound and westbound trains, in the dynamic headway approach the average delay for the northbound and southbound trains is given more weight than the eastbound and westbound trains in allocating the track resources at the junction. Therefore the dynamic headway approach is trying to decrease the overall delay, not just delay for the trains traveling eastbound and westbound. However, as the congestion increases the dynamic headway approach begins to significantly outperform the constant headway approach for the eastbound and westbound trains.

6. Conclusion

In summary, we considered the dynamic headway control problem for scheduling of heterogeneous trains through a shared railway network. A mathematical programming formulation is presented. The nonlinear and integral nature of the formulation makes it too complex to solve the problem directly. Therefore heuristic approaches are proposed to find an approximate solution. Traditional simulation approaches of the train movement model headway as fixed blocks which cannot be used to model dynamic headway. Hence, we present a new simulation modelling approach to represent dynamic headway control. Simulation experiments show that the proposed heuristics for dynamic headway control outperform traditional constant headway approaches in terms of reducing overall delay.

In this study we assumed a simple greedy routing approach (i.e., route to the next node with the highest speed limit), future research can focus on more advanced algorithms for the routing decision. Also, although it is well known that scheduling multiple trains through the railway network is a NP-hard problem, it remains unknown whether scheduling a single train through an empty network under the dynamic headway setting is NP-hard or not.

7. Acknowledgement

This research was partially supported by the Volvo Research and Education Foundation.
8. References


