Simultaneously re-optimizing timetables and platform schedules under planned track maintenance for a high-speed railway network

Zhang, Qin; Lusby, Richard Martin; Shang, Pan; Zhu, Xiaoning

Published in:
Transportation Research Part C: Emerging Technologies

Link to article, DOI:
10.1016/j.trc.2020.102823

Publication date:
2020

Document Version
Peer reviewed version

Citation (APA):
Simultaneously re-optimizing timetables and platform schedules under planned track maintenance for a high-speed railway network

Qin Zhang\textsuperscript{a}, Richard Martin Lusby\textsuperscript{b,*}, Pan Shang\textsuperscript{a} and Xiaoning Zhu\textsuperscript{a}

\textsuperscript{a}School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
\textsuperscript{b}Department of Management Engineering, Technical University of Denmark, Kgs. Lyngby 2800, Denmark

ARTICLE INFO

Keywords:
Railway optimization
Flexible track utilization
Lagrangian Relaxation
Alternating Direction Method of Multipliers

ABSTRACT

Train timetabling and train platforming are problems of crucial importance when scheduling high-speed trains. Often, these problems are solved separately and in sequence. It is also not uncommon for the problems to be further decomposed by direction since the use of tracks is usually direction specific in a high-speed network. In this paper, we consider the optimization problem of integrating re-timetabling and re-routing decisions within station areas for multiple stations when scheduled maintenance renders the existing, optimized schedules infeasible. We model the underlying problems using a space-time network on a mesoscopic level and propose a 0-1 binary integer programming model that can simultaneously modify the timings and routes of trains from different directions. Two different solution approaches are described. The first is a commonly used Lagrangian Relaxation (LR) approach, while the second utilizes the Alternating Direction Method of Multipliers (ADMM) concept. For both methods, a time-dependent dynamic programming approach is used to solve the resulting subproblems. A comparison of the two approaches on instances provided by the Chinese high-speed railway indicates that the ADMM-based approach provides tighter upper bounds and typically requires fewer iterations than the Lagrangian Relaxation approach. Furthermore, the results show that a flexible track utilization policy provides better timetables, with fewer cancellations and less total delay, than a fixed, dedicated direction, track policy.

1. Introduction

The Chinese high-speed railway has developed rapidly in the last few years. Its total length has grown to more than 35,000km, and it is now easily the largest high-speed railway network in the world. The scale and frequency of the system make the application of optimization based algorithms important when determining an efficient use of the available resources. Optimization algorithms are also important when re-optimizing the use of resources if adjustments are made to underlying schedules. Adjustments are frequently made to the Chinese high-speed railway timetables, typically as a result of increasing the capacity of busy lines, like the Beijing-Shanghai and Beijing-Guangzhou lines, but also as a result of maintenance activities which make parts of the network inaccessible. In such cases, it is important to be able to re-optimize the schedules without deviating too much from what was originally planned.

Optimizing the operations of a high-speed railway system is a challenging task. It is commonly divided into a number of different subproblems, as shown in Figure 1. These subproblems are addressed sequentially in most theoretical research, and in practice, due to their individual size and complexity. The solution to one problem is often used as fixed input to the next, and feedback loops can be used to revisit earlier planning problems. Based on the planning horizon, each of the subproblems is typically classified as a strategic, a tactical, or an operational level problem (Lusby et al., 2011). Strategic level problems focus on long term decision making such as network design and will make use of, among other things, future passenger flow forecasts. Tactical level problems, on the other hand, focus on the operation of the system and consider aspects such as timetable generation and the development of resource schedules. Decision making at these two levels is an offline problem. A high-speed railway is an open large-scale system that is easily influenced by various perturbations. Online problems are encountered on a day-to-day basis and must be solved quickly when optimized resource schedules must be revised due to an unforeseen event, termed a disruption, that prevents the schedules from being operated as planned. In this paper, we consider simultaneously...
re-optimizing the Train Timetabling Problem (TTP) and the Train Platforming Problem (TPP) for a network of stations to account for planned maintenance activities, which make areas of the network unavailable for extended periods of time. This is an offline problem; the planned nature of the maintenance makes this a tactical level problem.

The TTP optimizes the arrival and departure times of trains at each station while satisfying specific operational regulations, such as headway requirements and minimum required station dwell times. A timetable can be categorized as either cyclic or acyclic. A cyclic timetable repeats with a certain periodicity, e.g., every hour or every two hours. We consider acyclic timetabling since this is what is applied at the Chinese high-speed railway. As seen from Figure 1, the timetable coordinates several different subproblems. Depending on the scope of the timetabling problem, different infrastructure perspectives are considered when it is optimized. If a network-wide timetable is sought, then the macroscopic level is often considered. At this level, detailed station layouts are ignored and only a maximum number of trains that can be simultaneously present in each station is enforced, see e.g., Caprara et al. (2007). The track connections between two stations are modelled on an aggregated level, and trains from different directions are scheduled independently (Caprara et al., 2002; Zhang et al., 2019d). To determine an operationally feasible timetable, a microscopic network perspective must be considered. At this level, the movements of trains must be monitored on an individual block section level to guarantee train movements are conflict-free and respect infrastructure resource capacities (Zhang et al., 2019c). Block sections are sections of track that can accommodate at most one train at any given time. From a modelling perspective, mathematical formulations for timetabling on a microscopic level are prohibitively large for big networks. Since there is no guarantee that timetables generated on the macroscopic level are operationally feasible on a microscopic level, one possibility, and the approach adopted in this paper, is to consider a mesoscopic network perspective. This level captures aspects from both the macroscopic and microscopic levels. In particular, stations are considered from a microscopic level, while the track connections are modelled on an aggregated level.

The TPP primarily involves routing trains to a specific platform along conflict-free routes within station areas and typically follows the train timetabling phase. This problem is still often solved manually in practice (Cacchiani et al., 2016). The TTP and the TPP are highly interdependent and ideally should be simultaneously optimized. If a sequential approach is adopted, then the TPP must be solved for each station in the network. There is no guarantee that a feasible solution to the platforming problem exists for a given station when constrained to the times found by the timetabling problem (Zhang et al., 2019b). If just one TPP is infeasible, the TTP must be resolved. Such an iterative procedure must continue until feasibility is obtained for all platforming problems. Iterative procedures may take a long time and may result in poor quality feasible solutions. We, therefore, focus on developing an approach that integrates both problems from a mesoscopic perspective to mitigate these risks.

Timetables and platform schedules are susceptible to unplanned disruptions such as weather conditions, signalling system failures, and/or track malfunctions, but also to planned disruptions that appear in the form of requested mainte-
Simultaneously re-optimizing timetables and platform schedules

Incorporate possessions for certain tracks. Railway infrastructure must be kept in good condition to ensure safe and reliable operations. Infrastructure managers generate several preventive maintenance tasks to be executed in the near future. These preventive maintenance requests typically appear one day to several hours before the operation of the schedule and require planners to adjust schedules in such a way that they avoid the specified areas of track for several hours. This is what makes the problem that we consider an offline problem. Given the obvious interdependencies between timetabling and platforming, in this paper, we present an integrated approach for simultaneously re-optimizing these two problems given a set of requested maintenance possessions. Recently, we have witnessed an increase in the application of methodologies to solve integrated planning problems in the railway industry, see e.g., Meng and Zhou (2019); Burggraewe et al. (2017); however, to the best of our knowledge, we are not aware of any approaches that simultaneously re-optimize the timetable and platform schedule under planned track maintenance on a mesoscopic level. Track resources in the bottleneck areas of stations are often ignored at the train timetabling stage, and trains from different directions are usually treated independently (Yan et al., 2019), limiting the possibilities. This work, therefore, makes the following four contributions to the state-of-the-art within timetabling and platforming under track maintenance possessions.

1. A specific space-time network is developed to model the spatial and temporal requirements of the integrated problem. This permits the simultaneous construction of a feasible timetable and associated platform schedule. Multiple stations are modelled on a microscopic level. This allows us to capture network-wide impacts when adjusting the routings in one station.
2. A binary integer program is proposed to model the underlying optimization problem. The aim is to adjust the timetable and platform schedules in such a way that deviation from the planned schedules is minimized.
3. For the integrated problem, we compare a Lagrangian Relaxation (LR) solution with an approach based on the Alternating Direction Method of Multipliers (ADMM). The results indicate that the ADMM approach provides tighter upper bounds and typically uses fewer iterations than the LR approach. Where relevant, we also provide a comparison with the results obtained if a commercial solver applied to the full problem.
4. We test and assess the impact of a flexible station track utilization policy. Such a policy creates more flexibility when re-timetabling and re-platforming, but naturally creates larger optimization problems. We do not separate trains by direction, and any arrival or departure track can be allocated to any train if the station layout permits this. The majority of existing literature on the train timetabling problem at the tactical level does not consider this possibility. In practice, it is worthwhile considering this option, particularly if station capacity is reduced due to maintenance activities.

The remainder of this paper is organized as follows. Section 2 gives a comprehensive literature review on train timetabling and platforming problems, integrated optimization, and several possible solution methods. In Section 3, we illustrate the physical railway network and introduce its associated space-time network representation. We discuss how we model incompatibilities between trains and present the proposed binary integer program for modelling the integrated problem. Section 4 introduces the Lagrangian relaxation and ADMM based solution approaches, and the performance of these methods is analysed on several real-world instances in Section 5. Conclusions and future research directions are described in Section 6.

2. Literature review

The TTP is a classical optimization problem that has attracted considerable attention in the research literature. Caprara et al. (2002) highlighted its difficulty by reducing it to the Maximum Independent Set Problem. A train service plan which states global routes, stopping patterns and train frequencies (the number of times a certain train service runs within a given time period) is obtained from the Line Planning Problem (LPP) and is used as input to the TTP, see e.g., Qi et al. (2016); Yang et al. (2016). When adjusting an optimized timetable at the tactical level, the ideal timetable, or original timetable is assumed to be available (Caprara et al., 2002; Burdett and Koza, 2009). In general, a number of possible objective functions can be considered when adjusting a timetable. From a train operator perspective, these include maximizing train profit or maximizing the number of additional trains while minimizing the deviation from the original timetable. Passenger centered objectives focus on quality of service aspects such as minimizing the number of cancelled trains and the number of missed connections. Related literature can be classified according to microscopic and macroscopic perspectives. At the macroscopic level, Mixed Integer Linear Programs (MILPs), specifically big-M methods, are used to model "either-or" headway constraints and to decide the sequence of trains at stations (Zhou and
Simultaneously re-optimizing timetables and platform schedules

Zhong, 2005, 2007; Higgins et al., 1996; Carey, 1994a, b; Sotskov and Gholami, 2012; Veeleuturf et al., 2015). A space-time network, or space-time-state network, is often used to model the network possibilities. Binary decision variables can be used to model the selection of paths or arcs in such a time-discretized network. We refer readers to Cacchiani et al. (2008, 2016); Caprara et al. (2002); Meng and Zhou (2019) for further details. Zhang et al. (2019d) proposed a binary integer program to generate a cyclic timetable at the macroscopic level for trains running on a double-track railway corridor. The authors, as we do, developed and compared an LR method with an ADMM approach. The main difference between Zhang et al. (2019d) and this paper is that we design a new space-time representation to model acyclic timetabling at the mesoscopic level in order to more accurately capture infrastructure details. In addition, we also simultaneously schedule trains from different directions and assess the impact of a flexible track utilization policy.

A railway line is usually divided into several block sections at the microscopic level. MILPs based on a space-time network can still be applied, such as the work by Meng and Zhou (2014). Alternative graph formulations are common at this level. This approach models the train timetabling problem as a no-store multi-commodity scheduling problem (Mascis and Pacciarelli, 2002). To address the difficulty of solving the TTP, a variety of methods have been proposed. Corman et al. (2010) re-routed and re-timed trains under disruption with the help of a Tabu search approach that has several routing neighborhoods and new search strategies based on the local search algorithm of D’Ariano et al. (2008). D’Ariano et al. (2007) developed a Branch-and-Bound algorithm with several dynamic and static implication rules to identify the relationship between arcs during the procedure so that the near-optimal results can be obtained within short time limits. Zhou and Zhong (2005) solved a multiobjective high-speed railway scheduling problem with a Branch-and-Bound approach that used a beam search algorithm to generate pareto optimal solutions. A column generation approach, that combined heuristic and exact algorithms, was proposed in Cacchiani et al. (2008) to solve the TTP on a corridor. For periodic timetabling, Sparing and Goverde (2017) developed a preprocessing technique that removed redundant infrastructure constraints and introduced symmetry-breaking constraints to reduce the search space in order to be able to quickly generate large, stable timetables. Zhou and Teng (2016) and Meng and Zhou (2014) reformulated the TTP and designed an LR-based method to separate the large-scale problem into several subproblems on the macroscopic and microscopic levels, respectively. Other method, such as the direct application of commercial solvers, Zhang et al. (2019a); Gao et al. (2018), and genetic algorithm algorithms, Higgins et al. (1997), have also been applied.

The TTP has also been widely studied in the literature and applied to numerous real-world problems. The solution of the TTP typically provides the necessary input to the TPP at the macroscopic level in various theoretical studies and in practice. An example of this is the decision support system DONS that is used by the Dutch railway company NS. (Zwaneveld et al., 1996, 2001). Lusby et al. (2011) provided an extensive literature review on methods for the railway track allocation problem and highlighted the differences in how train route conflicts can be handled. The most common approach is to model the conflicts via a conflict graph and select conflict-free routes using a node packing formulation of Zwaneveld et al. (1996) or graph coloring type formulation, as described in Cardillo and Mione (1998). Delorme et al. (2004) proposed a Greedy Randomized Adaptative Search Procedure to generate upper and lower bounds and performed a small computational study to find the best configuration. Caprara et al. (2011) proposed a pattern incompatibility graph and designed a branch-and-price-and-cut method. Billionnet (2003) built two binary integer programs based on the work of Cardillo and Mione (1998) and directly used a commercial integer-programming solver to solve them. The disadvantage with graph coloring/node packing formulations is that they do not scale well. Significant preprocessing techniques are usually needed to produce a tractable model that can be solved directly with a commercial solver. Other approaches that have been proposed include constraint programming methods, multi-commodity flow formulations, and tailored heuristic algorithms. We refer the reader to Carey and Carville (2003); Rodriguez (2007); Fuchsberger and Lüthi (2007) for more details. Corman et al. (2009) considered the rescheduling variant of updating station train routes when delays occur and compared two alternative graph formulations, one on a disaggregated infrastructure level and one on an aggregated infrastructure level. The latter was shown to provide a good approximation of the optimal solution to the former. When re-adjusting platform schedules it is, however, important to simultaneously consider multiple stations from a delay propagation perspective.

Compared to a sequential approach, integrating two or more planning problems provides the possibility of finding improved solutions and has been a popular research direction of late. Common integrated approaches focus on combining the LPP and the TTP or look at combining the TTP and the Rolling Stock Planning Problem (RSPP). Burggraefe et al. (2017) constructed two modules and iteratively optimized the LPP and the TTP to improve the robustness of the timetable. Meng and Zhou (2019) integrated passenger demand, train timetabling, and rolling stock capacity considerations in a layered network and showed that integration improved the solution by 9% ~16%. Zhu et al. (2017)
Simultaneously re-optimizing timetables and platform schedules

and Wang et al. (2016) respectively proposed a two-stage genetic algorithm and a bi-level approach to jointly solve the LPP and the TTP for an urban rail line. Mixed integer linear programming techniques have been widely used to integrate rolling stock scheduling and train timetabling, such as the work by Xu et al. (2019); Wang et al. (2017); Walker et al. (2005). Laporte et al. (2017) used a multiobjective program to integrate passenger routing in the TTP and RSPP and analyzed the trade-off between operator and passenger requirements. Simultaneously optimizing the LPP and the Network Planning Problem (NPP) was investigated by Canca et al. (2019) and López-Ramos et al. (2017). Zeng et al. (2018) integrated the RSPP and the Crew Planning Problem (CPP) for a real-time disruption management process. The results indicated that integrated optimization provides better solutions than solving the individual problems in sequence. We summarize the types of integration and solution approaches in Table 1. Cacchiani et al. (2015) includes a general introduction to the non-periodic TTP and the TPP separately and emphasized that these two stages are very closely connected and should be solved simultaneously.

The TTP and the TPP are both NP-hard problems. Consequently, a number of decomposition-based solution methods have been proposed to solve them. Cacchiani et al. (2008) proposed a Dantzig-Wolfe reformulation and solved it via column generation. LR is a popular technique and works by relaxing a set of so-called coupling constraints, whose violation is penalized in the objective function. Such an approach results in several easy to solve subproblems, see e.g., Meng and Zhou (2014, 2019); Zhou and Teng (2016); Caprara et al. (2002). Cacchiani et al. (2015) compared the performance of LR-based approaches with exact branch-and-cut-and-price algorithms, and the results showed that the LR-based approaches obtained good solutions, while the latter struggled to even find a feasible solution for the large test cases. However, Niu et al. (2018) discussed the symmetry issues inherent for the vehicle routing problem and how a conventional LR method struggles to deal with it. A reliable dual decomposition variant based on ADMM was studied in Yao et al. (2019) to solve a vehicle routing problem. The authors proved this method could overcome the symmetry of a LR approach and could obtain better upper bounds. Due to the size and complexity of the integrated problem, we will design an ADMM-based solution framework to decompose the problem into a set of easier train-specific subproblems.

<table>
<thead>
<tr>
<th>Author</th>
<th>Problems</th>
<th>Solution algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canca et al. (2019)</td>
<td>NPP and LPP</td>
<td>Adaptive large neighborhood search</td>
</tr>
<tr>
<td>López-Ramos et al. (2017)</td>
<td>NPP and LPP</td>
<td>Corridor generation and line splitting</td>
</tr>
<tr>
<td>Zeng et al. (2018)</td>
<td>RSPP and CPP</td>
<td>Ant colony</td>
</tr>
<tr>
<td>Xu et al. (2019)</td>
<td>RSPP and TTP</td>
<td>Local branching and neighborhood search</td>
</tr>
<tr>
<td>Wang et al. (2017)</td>
<td>RSPP and TTP</td>
<td>CPLEX solver</td>
</tr>
<tr>
<td>Walker et al. (2005)</td>
<td>RSPP and TTP</td>
<td>Branch-and-Bound-based</td>
</tr>
<tr>
<td>Laporte et al. (2017)</td>
<td>RSPP and TTP</td>
<td>$\varepsilon$-constraint solution</td>
</tr>
<tr>
<td>Wang et al. (2016)</td>
<td>LPP and TTP</td>
<td>Bi-level approach</td>
</tr>
<tr>
<td>Zhu et al. (2017)</td>
<td>LPP and TTP</td>
<td>Two-stage genetic</td>
</tr>
<tr>
<td>Burggraeye et al. (2017)</td>
<td>LPP and TTP</td>
<td>Heuristic approach</td>
</tr>
<tr>
<td>Meng and Zhou (2019)</td>
<td>LPP and TTP</td>
<td>Lagrangian relaxation</td>
</tr>
<tr>
<td>This paper</td>
<td>TTP and TPP</td>
<td>ADMM</td>
</tr>
</tbody>
</table>

Several studies address the integration of train schedules and preventive maintenance planning from a microscopic perspective. Luan et al. (2017) formulated a MILP based on network cumulative flow variables to optimize the integrated problem. Maintenance tasks, which had flexible start times, were modelled as virtual trains. The objective function minimized the total train travel time deviation from the ideal timetable without any cancellations. An LR-based approach was used to solve the model. Zhang et al. (2019c) investigated simultaneously optimizing train timetabling and track maintenance task scheduling. A large MILP that minimized the total travel time of all trains plus a maintenance tardiness cost was proposed. The authors devised a heuristic algorithm that iteratively fixes train routes and maintenance tasks. D’Ariano et al. (2019) developed a bi-objective MILP and explored the impact of keeping or modifying train routes that were specified in the original timetable. This is different from the flexible track utilization strategy that we consider. Arenas et al. (2018) proposed a MILP formulation that minimized a weighted sum of arrival delays at the train’s destination and intermediate stops given a pre-defined maintenance task, with respect to the original timetable. Research that considers the TTP with maintenance activities from a macroscopic perspective can also
Simultaneously re-optimizing timetables and platform schedules

be found. Bahramian and Bagheri (2015) proposed a simulation-based approach for solving the problem given pre-defined periodic track maintenance tasks. Caprara et al. (2006) solved a train scheduling problem that considered track segment maintenance within a specified planning period. A heuristic LR approach was applied to obtain a solution. Table 2 provides a summary of related research on the topic of integrating train timetabling and track maintenance scheduling.

Table 2
Comparison of works on the integration of train timetabling and maintenance

<table>
<thead>
<tr>
<th>Author</th>
<th>Perspective</th>
<th>Solution algorithm</th>
<th>Modelling approach</th>
<th>TPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luan et al. (2017)</td>
<td>Micro</td>
<td>LR</td>
<td>Space-time network</td>
<td>Implicit</td>
</tr>
<tr>
<td>Zhang et al. (2019c)</td>
<td>Micro</td>
<td>Heuristic</td>
<td>Big-M formulation</td>
<td>Implicit</td>
</tr>
<tr>
<td>D’Ariano et al. (2019)</td>
<td>Micro</td>
<td>Pareto optimal method</td>
<td>Big-M formulation</td>
<td>Implicit</td>
</tr>
<tr>
<td>Arenas et al. (2018)</td>
<td>Micro</td>
<td>MILP solver&amp; Two-phase method&amp; Heuristic</td>
<td>Big-M formulation</td>
<td>Implicit</td>
</tr>
<tr>
<td>Bahramian and Bagheri (2015)</td>
<td>Macro</td>
<td>Simulation-based method</td>
<td>Simulation</td>
<td>Ignored</td>
</tr>
<tr>
<td>Caprara et al. (2006)</td>
<td>Macro</td>
<td>LR</td>
<td>Space-time network</td>
<td>Ignored</td>
</tr>
<tr>
<td>Van Aken et al. (2017)</td>
<td>Macro</td>
<td>CPLEX</td>
<td>PESP formulation</td>
<td>Ignored</td>
</tr>
<tr>
<td>This paper</td>
<td>Meso</td>
<td>LR&amp;ADMM</td>
<td>Space-time network</td>
<td>Explicit</td>
</tr>
</tbody>
</table>

3. Problem description and formulations

This section describes the problem in more detail and presents the proposed mathematical model. We discuss how we model the physical network and the space-time networks in Sections 3.1 and 3.2, respectively. As they are central to the mathematical model, a detailed description of incompatible arc sets is provided in Section 3.3. The full mathematical model is given in Section 3.4.

3.1. Railway network description and representation

High-speed railway networks are divided into two parts: sections (or segments) and stations. Sections are normally two-way double-track lines that consist of several block sections. A block section can only be occupied by at most one train at each time instant. A station consists of tracks and is comprised of a bottleneck area and a platform area, see Figure 2. Arrival and departure tracks in the platform area handle arrivals and departures and can accommodate any high-speed train, regardless of length, in China. Trains interact with each other in the the bottleneck areas and it is not permitted for a train to stop in this region. The layouts of the high-speed railway station in different countries have different properties. For example, in Europe, a platform may be divided into several sub platforms that can simultaneously accommodate different trains and allow for (de)coupling procedures. In the Chinese high-speed railway, a platform normally only serves one train at a time, and (de)coupling procedures at a platform do not typically occur.

As can be seen from Figure 2, high-speed railway stations in China are characterized by its common symmetric layout. For a railway corridor, stations are numbered. Outbound trains run in the direction of increasing station number, while inbound trains run in the opposite direction. Section tracks generally serve trains for the corresponding direction. Arrival and departure tracks within a station can be classified as platform tracks which can be used by trains that will stop at the station, and main lines, which can be used by trains that just pass through the station. Both of these can be further divided into outbound and inbound parts, according to the corresponding train direction. There are two station track utilization policies: fixed and flexible. For both policies (fixed or flexible), any main lines in the station can only be traversed by trains from the same direction. Platform tracks can only be occupied by trains from the same direction under a fixed track utilization rule. Such a policy allows the whole railway line to be divided into outbound and inbound parts, which could then be optimized separately. A flexible track utilization policy, however, allows trains to be assigned to any platform track in the station. In this case, trains from different directions have the potential to interfere with each other in the bottleneck area, and consequently all trains should be optimized simultaneously. Most current literature adopts a fixed track utilization policy when timetabling problem and/or platforming trains, see e.g., Zhang et al. (2020); Caprara et al. (2006). In this paper, we will investigate the benefits of a flexible policy. To illustrate the difference, for each train, arrival and departure tracks in the same direction are defined as normal, while those in the opposite direction will be defined as reverse.
Simultaneously re-optimizing timetables and platform schedules

Figure 2: Network representations from the timetabling and platforming perspectives.

The outermost home signal defines the boundary of the station. Trains occupy certain station track resources through different kinds of routes. Routes are classified as receiving (or arrival), departure, or through routes. A through route is traversed by a nonstop train and combines one receiving route and one departure route. Trains on a through route do not dwell at the station. Trains that stop at the station arrive via a receiving route, stop at a platform track to allow passengers on and off, and then leave the station via a departure route. Note that nonstop trains in the original timetable might not necessarily be nonstop trains in the revised timetable. However, the opposite case is not possible; trains cannot skip designated stops in the revised timetable. No train is permitted to stop on the mainlines of a station. In practice, the receiving route starts from the outermost home signal and ends at the starting signal, while the departure route starts from a starting signal and ends at the outermost reverse home signal (i.e., the home signal for trains arriving at the station on the reverse section track). However, trains will never pass a starting signal before leaving. To keep the continuity of train trajectory, we introduce stop points as the end of a receiving route and as the start of a departure route. Without loss of generalization, we separate one through route into a receiving route and a departure route with virtual stop points. Accordingly, for each train, receiving and departure routes can be further classified into two parts: normal and reverse. Normal routes assign trains to the normal arrival-departure tracks and reverse routes allow trains to use reverse arrival-departure tracks. For instance, receiving route 1 and departure route 1 in Figure 2 are normal routes and receiving route 2 and departure route 2 are reverse routes for outbound trains.

We consider a rail corridor that connects multiple stations. Furthermore, we assume that there is one single track in each direction between two successive stations and that no infrastructure crosses this network. In some high-speed railway stations, such as hubs that connect multiple lines, there are many routing possibilities between one boundary node and one (virtual) stop point; however, one basic route is typically used in practice, (Zhang et al., 2019b). Furthermore, we do not have to consider routing restrictions that might be induced due to required shunting operations, since the (de)coupling of rail cars at Chinese high-speed railway stations is uncommon. Station tracks can be shared
Simultaneously re-optimizing timetables and platform schedules

by outbound and inbound trains if the station layout allows for this. The physical railway network can be represented as different kinds of nodes and arcs between the nodes. The (virtual) stop points on the arrival-departure tracks and boundaries of the stations are defined as nodes and the track segment between any two nodes is defined as an arc. The node set that we use to model the physical railway network can be defined as

\[ N = N^a \cup N^d \cup N^s, \]

where \( N^a, N^d \) and \( N^s \) represent the sets of arrival and departure boundaries, and (virtual) stop points, respectively. We denote the set of stations on the corridor by

\[ K = \{1, \ldots, [K]\}, \]

where \([K]\) is the total number of stations. For each station \( k \in K \), its node set is given by \( N_k \) and has the form \( \cup N^a_k \cup N^d_k \cup N^s_k \). Note that \( N^a_k \) and \( N^d_k \) can be further classified into outbound and inbound nodes. Arcs are classified into station arcs and section arcs. Section arcs have an associated direction while station arcs depend on the relevant directions and the track utilization policy. According to the types of routes, station arcs can be divided into receiving arcs and departure arcs. Figure 3 gives a simple example of how we model a physical network.

We consider a set of maintenance possessions. Each possession is a task associated with a particular infrastructure resource and has a fixed start and a fixed end time. Each preventive maintenance task hence claims a track resource for a specified interval of time, and this reduces the capacity of the network. In this paper, we consider two types of maintenance tasks: station arrival-departure track maintenance and track segment maintenance. Arcs of the space-time network corresponding to resources that are unavailable due to maintenance can be removed. When re-timetabling and re-platforming trains under track maintenance possessions, we permit the following modification strategies:

1. **Shift departure times**: Trains can have their departure time at their origin station delayed within a certain time window. Bringing forward trains is not allowed as passengers might miss their planned trains.
2. **Stretch dwell times**: All trains can have their dwell time at intermediate stations increased.
3. **Re-route**: Trains can be re-routed through stations to avoid problematic areas.
4. **Cancellations**: If it is not possible to find a feasible solution when adjusting times and platforms, then it is possible to cancel a train.

### 3.2. Space-time railway network representation

The aperiodic TTP can be modelled using a directed, acyclic, space-time graph. Time is typically discretized into minutes for entire the planning horizon, which we denote as \( T \). Vertices in the network correspond to arrival and departure times at specific stations, see e.g. Caprara et al. (2002); Cacchiani et al. (2015); Zhang et al. (2019d). In this paper, we introduce more different types of vertices and arcs in our space-time network \( G = (V, A) \) to model the occupation of resources. Each node in the physical railway network is extended to a set of vertices associated with time instants within a planning horizon \( T \). The vertex set \( V \) has the form \( \cup V^a \cup V^d \cup V^s \). A vertex \( v \in V \) can be represented as a pair \((i, t)\) where \( i \in N \). The vertex set available to train \( f \) is represented as \( V_f = \cup V^a_f \cup V^d_f \cup V^s_f \). The running time or waiting time on a certain physical arc is illustrated by the space-time network arc \((i, j, t, \tau)\) upstream from the vertex \((i, t)\) and downstream to the vertex \((j, \tau)\). The arc set \( A \) of the whole space-time network consists of:

![Figure 3: An example of the physical railway line representation.](image-url)
Simultaneously re-optimizing timetables and platform schedules

Figure 4: An example of virtual origin and destination waiting arcs.

1) Virtual origin waiting arcs $A^{vo}$ and destination waiting arcs $A^{vd}$

Virtual waiting arcs are constructed from the actual origin and destination nodes of all trains along the time dimension. Unlike the studies of Cacchiani et al. (2015, 2016), we introduce virtual waiting arcs at the actual origin nodes and destination nodes to represent the source and sink arcs. We define the origin and destination node sets as $N^o$ and $N^e$, respectively. Virtual waiting arcs indicate the time shift at the origin and destination nodes from the original plan for each train. For a physical origin or destination node $i \in N^o \cup N^e$, the corresponding virtual waiting arcs are represented as $(i, i, t, t + 1)$ from vertex $(i, t)$ to vertex $(i, t + 1)$, for $t = 1, \ldots, T - 1$. Figure 4 gives an example of these two kinds of virtual waiting arcs. Nodes 1 and 8 are origin nodes and nodes 2 and 7 are destination nodes for outbound and inbound trains, respectively. Virtual origin waiting arcs $(1, 1, t, t + 1)$ and $(8, 8, t, t + 1)$ and virtual destination waiting arcs $(2, 2, t, t + 1)$ and $(7, 7, t, t + 1)$, for $t = 1, \ldots, T - 1$ are constructed. A train’s origin node refers to the arrival boundary that connects the considered railway network with the part of the network not considered or a depot (where idle trains wait). In practice, a train cannot dwell at the origin node and an increase in dwell time here will propagate the delay outside the considered area, and we want to resolve the delay in our considered area as much as possible. We, therefore, set the cost of a virtual origin waiting arc to $a_1$, where $a_1 > 1$. Waiting at a destination node indicates that a train has already left the system. The cost of a virtual destination waiting arc is therefore set to zero.

2) Station receiving arcs $A^a$ and station departure arcs $A^d$

These two types of station arc sets are constructed from physical station receiving and departure arcs and describe the receiving and departure routes associated with the time instants. As we mentioned in Section 3.1, we allow a flexible track utilization policy. When a platform tracks in one direction is under maintenance, trains from this direction can use the reverse platform tracks. We discourage the use of reverse routes in the normal situation due to the high possibility of creating conflicts with other trains in the bottleneck areas. The costs of reverse receiving and departure arcs are therefore higher than the normal arcs. For each physical receiving (departure) arc $(i, j)$, its corresponding space-time receiving (departure) arc is represented as $(i, j, t, t + r_{ij})$, for $t = 1, \ldots, T - r_{ij}$, where $r_{ij}$ is the running time on physical arc $(i, j)$. Figure 5 shows the receiving and departure arcs in Station 1 (from Figure 3) for trains from two directions, respectively. The difference in train running times on different routes in a certain station is very small. For a station with 17 arrival-departure tracks in China, the running times of trains on different routes range from 1min20s to 2min10s. In this paper, we assume the running times of different routes in one station are the same. Accordingly, the receiving and departure arc can also be classified into normal and reverse. For each train, the costs of receiving...
Simultaneously re-optimizing timetables and platform schedules

Figure 5: Illustration of space-time receiving and departure arcs.

Figure 6: Representation of station waiting arcs.

and departure arcs \((i, j, t, t + r_{ij})\) are calculated as:

\[ c_{f}^{i,j,t,t+r_{ij}} = \begin{cases} r_{ij}, & \text{if } (i, j) \text{ is a normal physical arc for train } f \\ \alpha_{2} \cdot r_{ij}, & \text{if } (i, j) \text{ is a reverse physical arc for train } f \end{cases} \]  \hspace{1cm} (1)

where \(\alpha_{2} > 1\).

3) Station waiting arcs \(A^{st}\)

Station waiting arcs connect stop points along the time dimension and indicate the dwelling of trains on station platform tracks. As no trains can dwell on the main lines in a station, virtual stop points cannot be extended. We denote the set of stopping points in the physical network by \(N^{st}\). For each stop point \(i \in N^{st}\) at a given station, this type of arc connects vertex \((i, t)\) and \((i, t + 1)\), for \(t = 1, \ldots, T - 1\). An example is given in Figure 6. Waiting arcs \((i, i, t, t + 1)\) are added for stop points \(i = 3\) and \(i = 6\).

In each station, every train is scheduled to a certain station track in the ideal timetable. Considering this track prefer-
Simultaneously re-optimizing timetables and platform schedules

Figure 7: Illustration of section running arcs.

ence, the costs of waiting arcs for each train $f$ in each station $k$ are:

$$c_{i,j,t,t+1}^f = \begin{cases} 
\alpha_3, & \text{if } i \text{ is not the preferred track for train } f \\
\alpha_3 - 1, & \text{if } i \text{ is the preferred track for train } f 
\end{cases}$$

where $i$ denotes the platform for train $f$. Extra waiting time causes more delay, and this should be minimized. Therefore we set $\alpha_3 > \alpha_2$.

4) Section running arcs $A_{se}$

A section running arc models the movement of a train on a segment track between two stations. Each physical section arc $(i, j)$ has a set of space-time section running arcs, each of which is of the form $(i, j, t, t + r_{ij})$ for $t = 1, \ldots, T - r_{ij}$ and connects $(i, t)$ with $(j, t + r_{ij})$. We assume that the travel time $r_{ij}$ on segment $(i, j)$ is constant. Figure 7 illustrates the construction of section running arcs for the physical section arc $(7, 9)$ in Figure 3. In the example, the running time is four minutes, and the constructed arcs of the form $(7, 9, t, t + 4)$, for $t = 1, \ldots, T - 3$. The cost of a section running arc is equal to the running time.

The arc set of the full space-time network is $A = A_v^o \cup A_v^d \cup A_d \cup A_a^d \cup A_a^o \cup A_{se}$. Figure 8 gives an example of the full space-time network for two stations and one section from Figure 3. Note that not all arcs in the space-time network can be used by all trains. We use the notation $A_f = A_f^o \cup A_f^d \cup A_f^a \cup A_f^d \cup A_f^o \cup A_f^{se}$ to indicate the possible arc set for train $f \in F$.

3.3. Determining incompatible arc sets

A minimum safety distance, or headway, must be enforced between any two consecutive train movements that share infrastructure. This primarily happens on sections and within the bottleneck areas of the station. Headways prevent trains from getting too close to one another. These requirements imply that some arcs in the space-time network cannot be used simultaneously. In this paper, we define and precalculate incompatible arc sets for each section running arc and for every station receiving arc and departure arc.

3.3.1. Incompatible arc set for section running arcs

Minimum headway requirements for trains on section tracks can be enforced through minimum departure and arrival time headway requirements at the two stations involved, as shown in Figure 9. Here, $T_{i,j}^d$ denotes the headway time necessary between trains entering a section, while $T_{i,j}^a$ denotes the headway time necessary for trains leaving the section. For any section running arc, we can determine a priori a set of – incompatible – section running arcs that cannot be chosen simultaneously due to headway violations. For arc $(i, j, t, \tau) \in A_{se}$, the set of incompatible arcs includes itself and any later arcs for which the associated times violate the arrival or departure headway restriction. This set is defined as follows.

$$\psi_{i,j,t,\tau}^a = \{(i, j, t', \tau') | 0 \leq (t' - t) < T_{i,j}^d \cup 0 \leq (\tau' - \tau) < T_{i,j}^a\}, \forall (i, j, t, \tau) \in A_{se}$$

Figure 9 illustrates the headway requirements for the section running arc $(7, 9, 3, 7)$. Assuming a headway time of 2 minutes, the incompatible arc set for arc $(7, 9, 3, 7)$ is $\psi_{7,9,3,7}^a = \{(7, 9, 3, 7), (7, 9, 4, 8)\}$.

Qin Zhang et al.: Page 11 of 34
Simultaneously re-optimizing timetables and platform schedules

Figure 8: Illustration of the full space-time railway network.

Figure 9: Illustration of the incompatible arc set for a section running arc.

3.3.2. Incompatible arc set for station receiving and departure arcs

The tracks in the bottleneck area, where most conflicts occur, are more complex than the segment area. We therefore compute the incompatible arc set for receiving and departure arcs in two steps.

(1) Determine the incompatible arc set for each physical station arc from the microscopic perspective

Given the topology of a high-speed railway station from the microscopic perspective, we consider important resources
Simultaneously re-optimizing timetables and platform schedules

Figure 10: The topology of railway station from the microscopic perspective.

Figure 11: Illustration of the incompatible arc set for a space-time station receiving arc.

such as a crossing or a switch. On a microscopic level, any physical arc can be represented as a sequence of such resources. Any two physical arcs are assumed to be incompatible if they have at least one such resource in common.

We define an incompatible arc set \( \psi(i, j) \) for each physical station arc \((i, j)\). Figure 10 provides an example of this. The physical arc \((1, 3)\) has an associated set of resources \(\{A, B, C, D\}\) while arc \((4, 2)\)’s resource set is \(\{E, C, B, F, G\}\). Since these two arcs share the two resources \(B\) and \(C\), \(\psi(1, 3)\) contains arc \((4, 2)\) and \(\psi(4, 2)\) contains \((1, 3)\).

(2) Determine the incompatible arc set for each space-time station arc from the mesoscopic perspective

The incompatible arc set for each space-time station arc can be determined based on the incompatible arc set for each physical arc. The difference between the incompatible arc set for physical section arcs and the incompatible arc set for either physical station receiving arcs or departure arcs is that there can be multiple trains in one section as long as these trains satisfy the section time headway requirement. There can, however, only be one train on any physical station arc in an incompatible arc set at any given time instant. The headway requirement for two consecutive trains in the bottleneck area is \(T^r\). The incompatible arc set for a space-time receiving or departure arc \(\psi^{a}_{i,j,t,\tau}\) is defined to be the set of space-time arcs associated with the physical arcs in the conflicting set \(\psi(i, j)\) that do not satisfy the headway requirement. This is given as follows:

\[
\psi^{a}_{i,j,t,\tau} = \{(l, m, t', \tau') \mid t \leq t' < \tau + T^r, (l, m) \in \psi(i, j)\}, \forall (i, j, t, \tau) \in A^a \cup A^d.
\] (4)

For instance, as shown in Figure 11, the incompatible physical arc set \(\psi(1, 4) = \{(1, 3), (1, 4), (1, 6), (3, 2)\}\). If the headway must be at least one minute, i.e., \(T^r = 1\), then the incompatible arc set for arc \((1, 4, 3, 4)\) is

\[
\psi^{a}_{1,4,3,4} = \{(1, 3, 3, 4), (1, 4, 3, 4), (1, 6, 3, 4), (3, 2, 3, 4), (1, 3, 4, 5), (1, 4, 4, 5), (1, 6, 4, 5), (3, 2, 4, 5)\}.
\]

3.3.3. Arc elimination due to maintenance tasks

In a pre-processing step, for each maintenance task, the space-time arcs that are associated with the corresponding track resource and which conflict with the maintenance interval will be removed directly. As an example, consider
Simultaneously re-optimizing timetables and platform schedules

Figure 12: Elimination of space-time segment arcs.

Figure 12. The maintenance task for track segment \((i, j)\) starts from \(T_{ij}^0\) and ends at \(T_{ij}^e\). Clearly, space-time arcs in arc set

\[ \{(i, j, t, \tau)|t \in (T_{ij}^0 - r_{ij}, T_{ij}^e]\} \]

are incompatible with the maintenance task and will be eliminated from the train space-time network. Similarly, if a station arrival-departure track \(i\) undergoes a maintenance task from \(T_{ii}^0\) to \(T_{ii}^e\), any space-time arc associated with node \(i\) that conflicts with the maintenance interval, i.e.,

\[ \{(i', j', t, \tau)|i = i' \cup j = j', t \in (T_{ii}^0 - r_{i'j'}, T_{ii}^e]\} \]

will be removed.

3.4. Model formulation

When modelling the problem mathematically we make several assumptions. We explicitly state these here.

(1) An ideal timetable, infrastructure information, and the set of track maintenance tasks are available.
(2) Track segment traversal time is constant.
(3) The running time of trains on different routes within a station is assumed be constant. The speed limitation when a train crosses a switch in a Chinese high-speed railway station is normally 80km/h. As mentioned earlier, the running time difference of every route through the bottleneck area at a station is generally less than one minute.
(4) Trains only use the basic route between a boundary node and any arrival-departure track in a station. The number of possible routes connecting the station boundary and one station track may be more than one. However, normally only one basic route will be used in practice.
(5) Train length is not explicitly included in the model. It is, however, implicitly considered when calculating the running times of the track resources. This is consistent with similar previous studies (Meng and Zhou, 2014; Qi et al., 2016; Zhang et al., 2020)
(6) All track sections that comprise a station route are released simultaneously. This is because we model track resources in the bottleneck area of a station on an aggregated level. As such, it is not possible to model a track section by track section release system without increasing the size of the model substantially. Corman et al. (2009) show that there is generally not a big difference between the two and that enforcing a route-release system provides a good approximation of the optimal solution that considers a sectional-release system. This is sufficient for the tactical level problem we consider.
(7) Time is discretized into one minute intervals.
Simultaneously re-optimizing timetables and platform schedules

3.4.1. Notation

As mentioned earlier, for the problem at hand we assume the following information is available: the topology of the railway network, a reference timetable, the scheduled maintenance activities, and any station specific data. We also assume that all headway times (on sections, station routes, and between movements on the same arrival-departure tracks), the planning horizon, the allowed time shifts and time stretches, and the bounds on dwell times have been specified. The timetable contains the planned activities for all planned trains. This includes the running direction, the station sequence, the stopping pattern, the minimum dwell times, preferred station routes, and the planned arrival and departure times at each station. Combining this information with the railway network topology and station data, the scheduled maintenance windows, and the considered time horizon, all possible arcs in the space-time network can be generated. A summary of the notation used, along with some parameters, is given in Tables 3, 4 and 5.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Definition of indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation</td>
<td>Definition</td>
</tr>
<tr>
<td>$i, j, i', j', i'', j''$</td>
<td>Physical node index</td>
</tr>
<tr>
<td>$(i, j)$</td>
<td>Physical arc index</td>
</tr>
<tr>
<td>$i, i', i'', j, j', j''$</td>
<td>Time index, $= 1, \ldots, T$, $T$ is the planning time horizon</td>
</tr>
<tr>
<td>$(i, t)$</td>
<td>Space-time vertex index</td>
</tr>
<tr>
<td>$(i, j, t, \tau)$</td>
<td>Space-time arc index</td>
</tr>
<tr>
<td>$f, f'$</td>
<td>Train index</td>
</tr>
<tr>
<td>$k$</td>
<td>Station index</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Definition of sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation</td>
<td>Definition</td>
</tr>
<tr>
<td>$N$</td>
<td>Set of nodes</td>
</tr>
<tr>
<td>$K$</td>
<td>Set of stations</td>
</tr>
<tr>
<td>$N^a_k$</td>
<td>Set of home signal nodes (arrival boundary nodes) in station $k$</td>
</tr>
<tr>
<td>$N^d_k$</td>
<td>Set of reverse home signal nodes (departure boundary nodes) in station $k$</td>
</tr>
<tr>
<td>$N^s_k$</td>
<td>Set of (virtual) stopping points in station $k$</td>
</tr>
<tr>
<td>$N^t_k$</td>
<td>Set of stopping points in station $k$</td>
</tr>
<tr>
<td>$V$</td>
<td>Set of train space-time vertices</td>
</tr>
<tr>
<td>$A$</td>
<td>Set of train space-time arcs</td>
</tr>
<tr>
<td>$A^{se}$</td>
<td>Set of section arcs in the space-time network ($\subset A$)</td>
</tr>
<tr>
<td>$A^s, A^d$</td>
<td>Set of station receiving arcs and departure arcs in the space-time network ($\subset A$)</td>
</tr>
<tr>
<td>$A^{st}$</td>
<td>Set of station waiting arcs in the space-time network ($\subset A$)</td>
</tr>
<tr>
<td>$\delta^+_f, \delta^-_f$</td>
<td>Set of arcs that leaving from or entering to vertex $(i, t)$</td>
</tr>
<tr>
<td>$\psi_{i,j,t,\tau}$</td>
<td>Incompatible arc set for arc $(i, j, t, \tau)$</td>
</tr>
<tr>
<td>$\psi(i, j)$</td>
<td>Incompatible arc set for physical arc $(i, j)$</td>
</tr>
<tr>
<td>$F$</td>
<td>Set of trains, index by $f, f'$, i.e., $f, f' \in F$</td>
</tr>
<tr>
<td>$V_f$</td>
<td>Set of possible vertices for train $f \in F$</td>
</tr>
<tr>
<td>$A_f$</td>
<td>Set of possible arcs for train $f \in F$</td>
</tr>
<tr>
<td>$\sigma^+_f$</td>
<td>Set of arcs for train $f \in F$ emanating the origin node at the possible earliest time</td>
</tr>
<tr>
<td>$K_f$</td>
<td>Set of stations that train $f$ travels through</td>
</tr>
</tbody>
</table>

To determine a new timetable and a corresponding set of train routes, we essentially have to route each train between their origin and destination nodes in their respective space-time networks. This involves selecting which of the arcs in the space-time network will be used. Therefore, we define the binary decision variables $x^f_{i,j,t,\tau}$ which indicate whether arc $(i, j, t, \tau)$ will be used by train $f \in F$ ($x^f_{i,j,t,\tau} = 1$) or not ($x^f_{i,j,t,\tau} = 0$). For notational simplicity, we denote an arc $(i, j, t, \tau)$ as $g$ and further introduce $o_g = i, d_g = j, \hat{o}_g = \tau$, and $\hat{d}_g = t$. We also permit trains to be cancelled and indicate this possibility with the binary decision variables $\theta_f$ for $f \in F$. The variable takes the value one if train...
Simultaneously re-optimizing timetables and platform schedules

Table 5
Definition of parameters

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k(i)$</td>
<td>Station $k$ that contains node $i$</td>
</tr>
<tr>
<td>$T'_k$</td>
<td>Running time of receiving or departure route at station $k \in K$</td>
</tr>
<tr>
<td>$T'$</td>
<td>Headway for occupation of station receiving and departure routes</td>
</tr>
<tr>
<td>$T^{d}_{i,j}$</td>
<td>Headway time for trains entering the same section arc</td>
</tr>
<tr>
<td>$T^{l}_{i,j}$</td>
<td>Headway time for trains leaving the same section arc</td>
</tr>
<tr>
<td>$T^{i}_{i}$</td>
<td>Track usage headway for two consecutive trains at stopping point $i$</td>
</tr>
<tr>
<td>$\beta_{f}$</td>
<td>Earliest origin vertex and latest destination vertex for train $f \in F$</td>
</tr>
<tr>
<td>$c^f_{i,j,t}$</td>
<td>Fixed cost of arc $(i,j,t)$ for train $f \in F$</td>
</tr>
<tr>
<td>$\hat{c}^f_{i,j,t}$</td>
<td>Modified cost of arc $(i,j,t)$, including fixed cost and Lagrange cost</td>
</tr>
<tr>
<td>$\tilde{c}^f_{i,j,t}$</td>
<td>Modified cost of arc $(i,j,t)$ for train $f \in F$, including fixed cost, Lagrange cost and quadratic cost</td>
</tr>
<tr>
<td>$S^f_k$</td>
<td>Scheduled stopping pattern of train $f \in F$ at station $k \in K$, $=1$ if train stop at station $k$; $=0$ otherwise</td>
</tr>
<tr>
<td>$D^f_k$</td>
<td>Minimum dwell time of train $f \in F$ at station $k \in K$</td>
</tr>
<tr>
<td>$D^{\text{max}}_k$</td>
<td>Maximum dwell time of train $f \in F$ at station $k \in K$</td>
</tr>
<tr>
<td>$P^f_k$</td>
<td>Planned departure time for train $f \in F$ at station $k \in K$</td>
</tr>
<tr>
<td>$C$</td>
<td>Cost for canceling a train</td>
</tr>
<tr>
<td>$U_1, U_2$</td>
<td>A sufficiently large number</td>
</tr>
</tbody>
</table>

$f \in F$ is cancelled and is zero otherwise. The decision variables are stated in Table 6.

Table 6
Decision variables of the model

<table>
<thead>
<tr>
<th>Notations</th>
<th>Detailed definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^f_g \in {0, 1}$</td>
<td>Binary decision variable that indicates whether train $f \in F$ uses arc $g$ ($=1$) or not ($=0$)</td>
</tr>
<tr>
<td>$\theta^f_f \in {0, 1}$</td>
<td>Binary decision variable that indicates whether train $f \in F$ is cancelled ($=1$) or not ($=0$)</td>
</tr>
</tbody>
</table>

3.4.2. Space-Time Network Model (STMN)

The objective function for the TTP will typically minimize the operational cost, (Binder et al., 2017), minimize the total deviation time of all trains, (Meng and Zhou, 2014), or minimize the number of cancelled trains, (Veelenturf et al., 2015). In this paper, we minimize an objective that combines cancellation costs and weighted train travel times. The objective function is hence:

$$\min Z_1 = \sum_{f \in F} \left( C \cdot \theta^f_f + \sum_{g \in A_f} c^f_g \cdot x^f_g \right)$$

The following flow constraints must be satisfied:

$$\sum_{g \in \delta^+_f \cap A_f} x^f_g - \sum_{g \in \delta^-_f \cap A_f} x^f_g = 0 \quad \forall (o_g, \delta_g) \in V_f \setminus \{\beta_f, \gamma_f\}, \forall f \in F$$

Constraint (6) ensures that train $f \in F$ can use at most one outgoing arc from its origin node. Flow conservation to a destination node is then enforced through constraints (7). The following constraints enforce dwell time requirements:

Qin Zhang et al.: Page 16 of 34
Simultaneously re-optimizing timetables and platform schedules

Figure 13: Illustration of station routes.

\[
\sum_{g \in A_{st}^T} \sum_{o_g \in N_k^s} x_g^f \geq D_k^f \cdot (1 - \theta^f) \quad \forall f \in F, \forall k \in K_f \tag{8}
\]

\[
\sum_{(d_g, d_g', t', t'') \in A_f} \sum_{t' \in [d_g, d_g'] + D_g] \cup t' \neq d_g} x_g^f \leq 1 - x_g^i \quad \forall f \in F, \forall k \in K_f, \forall g \in A_f \cup \{o_g \neq d_g \cup d_g \in N_k^s\} \tag{9}
\]

\[
\sum_{g \in A_{st}^T} \sum_{o_g \in N_k^s} x_g^f \leq D_k^m \quad \forall f \in F, \forall k \in K_f \tag{10}
\]

Constraint (8) enforces the minimum required dwell time for a planned stop train, as long as the train is not cancelled. Constraint (9) states that if a planned nonstop train is assigned to platform track in station \(k\), i.e., \(x_g^f\) in the right-hand side is equal to 1, then this train must dwell for at least the minimum required dwell time of the platform track (i.e., departure arcs that result in less waiting time cannot be chosen). Recall that it is possible in the revised timetable to stop a previously planned nonstop train. The difference between constraints (8) and (9) is that the former enforces the minimum dwell time required by the planned timetable, while the latter enforces this for the revised timetable. Constraint (10) guarantees that train \(f \in F\) cannot dwell on a stopping track for longer than maximum permitted dwell time. A solution must also respect the following departure time constraint:

\[
\sum_{g \in A_{st}^T} \sum_{o_g \in N_k^s \cup t' \neq d_g} x_g^f \leq 1 - S_k^f \quad \forall f \in F, \forall k \in K_f \tag{11}
\]

In China, high-speed rail tickets are purchased well in advance of the trip, making adjustments to departure times somewhat difficult. Constraint (11) makes sure that the departures cannot leave the respective station earlier than the planned departure time.

Several different types of constraints are necessary to accurately enforce the headway requirement. To enforce headway times in the section and bottleneck area of a station, only one arc in any incompatible arc set can be selected. Constraint (12) enforces the headway requirement on the segment tracks.

\[
\sum_{(i', j', t', t'') \in A^s} \sum_{g \in A^s} x_g^f \leq 1 \quad \forall g \in A^s \tag{12}
\]

The headway requirements in the bottleneck area can be classified into three types, as shown in Figure 13. These are respectively, the headway between receiving routes, the headway between departure routes, and the headway between receiving and departure routes. We model these requirements as follows:

\[
\sum_{t' \in [d_g, d_g' + T_k^t + T_r]} \sum_{(o_g, j', t', t'') \in E_{st}^+} \sum_{f \in A_f} x_g^f \leq 1 \quad \forall (o_g, d_g) | o_g \in N_k^o, \forall k \in K \tag{13}
\]
Simultaneously re-optimizing timetables and platform schedules

**Figure 14:** An example of failure on platform track usage headway.

\[
\sum_{\tau' \in \{d_g, d_g + T' + T_r\}} \sum_{(i', d_g, d_g') \in \delta_{d_g, \tau'}} x_{i', d_g, d_g', \tau'}^f \leq 1 \quad \forall (d_g, d_g) \mid d_g \in N_k^d, \forall k \in K \quad (14)
\]

\[
\sum_{(i', j', t', \tau') \in U^d_g \setminus \{(i' = o_g, j' \neq d_g) \}} x_{i', j', t', \tau'}^f \leq 1 \quad \forall g \in A^d \quad (15)
\]

Constraint (13) enforces the headway requirement between two receiving routes. Similarly, Constraint (14) guarantees that the headway requirement is satisfied for two departure routes. Constraint (15) is similar to Constraint (12) and prevents violation of the headway between receiving and departure routes through the use of an incompatible arc set. As shown in Figure 13, Route 1 and Route 2 are arrival routes, and Route 3 is a departure route. Route 2 is incompatible with both Route 1 and Route 3. However, Route 1 and Route 3 can be performed by two trains simultaneously. Constraint (15) is defined for each receiving arc. The left-hand side excludes receiving arcs that have the same origin node but a different destination node from the incompatible arc set for the respective arc. Therefore, only receiving arcs corresponding to the considered arrival route and incompatible departure arcs are included in the left-hand side of Constraint (15). This guarantees the headway between arrival and departure routes.

The headway requirement between two consecutive trains that use the same arrival-departure track is more complicated to model. Enforcing station route headway requirements ensures that trains are sufficiently separated on main lines in a station. They do not, however, consider the stopping tracks. Figure 14 provides an example of this; train 2 will occupy the arc (3, 3, 5, 6) if no track usage headway requirement is enforced for the stopping track.

The difference between this type of headway constraint and the others is that we cannot determine the set of conflicting arcs for each of the waiting arcs a priori. As shown in Figure 14, the headway constraint on station routes ensures that arcs (1, 3, 3, 4) cannot be occupied by Train 2. However, it is possible for other trains to use arc (1, 3, 4, 5). In the example, the station route headway and station track headway are one minute and two minutes, respectively. In this case arc (3, 3, 5, 6) cannot be used by Train 2. However, this arc is allowed to be occupied by Train 1 if more dwell time is required. Without the station track headway constraint, Train 2 can use the light yellow route. This should not be possible. Therefore, we introduce an auxiliary binary decision variable \( y_{f,g} \) for each waiting arc \( g \in A^d \) to model this implicit occupation of track. This decision variable takes the value one if arc \( g \in A^d \) is implicitly occupied by train \( f \in F \) and is zero otherwise. Note that implicit occupation of track indicates that a train – does not – actually use the track, it just prevents other trains from doing so due to the headway requirement. The value of these new variables depends on the actual occupation of waiting arcs. Constraint (16) defines the relationship between the two types of occupation variables. The parameter \( l \) can be assigned any value larger than 1. Table 7 further elaborates on this relationship when \( T_f^r = 1 \). Constraint (17) enforces the headway requirement for stopping tracks; this constraint states that only one train can occupy a certain waiting arc, either implicitly or explicitly. The connection between explicit occupation variables and cancellation is given by constraints (18) and (19).

\[
\frac{\sum_{\delta_g - T_g^d \leq \tau' < \delta_g} x_{i', d_g, d_g', \tau'}^f \cdot \tau'}{U_1} - x_{g}^f \leq y_{f,g} < 1 - \frac{\sum_{\delta_g - T_g^d \leq \tau' < \delta_g} x_{i', d_g, d_g', \tau'}^f \cdot \tau'}{U_1} \quad \forall g \in A^d_f, \forall f \in F \quad (16)
\]
Simultaneously re-optimizing timetables and platform schedules

\[
\sum_{f \in F} \left( x_f^g + y_f^g \right) \leq 1 \quad \forall g \in A \quad (17)
\]

\[
\sum_{g \in A_f} x_f^g / U_2 + \theta_f \leq 1 \quad \forall f \in F \quad (18)
\]

\[
\sum_{g \in A_f} x_f^g + \theta_f \geq 1 \quad \forall f \in F \quad (19)
\]

Table 7
Relationship between two occupation variables for waiting arcs

<table>
<thead>
<tr>
<th>( x_{i,j,t-1}^f )</th>
<th>( x_{i,j,t+1}^f )</th>
<th>Value of the left side</th>
<th>Value of the right side</th>
<th>( y_{i,j,t+1}^f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>( 1 + \frac{1}{v_1} )</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>( \frac{1}{v_1} - 1 )</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Finally, the domains of the variables are given below.

\[
x_f^g \in \{0, 1\} \quad \forall f \in F, \forall g \in A \quad (20)
\]

\[
y_f^g \in \{0, 1\} \quad \forall f \in F, \forall g \in A^{st} \quad (21)
\]

\[
\theta_f \in \{0, 1\} \quad \forall f \in F \quad (22)
\]

The construction of the proposed space-time network automatically satisfies some constraints. These are:

1. Track uniqueness - trains can only occupy one arrival-departure track at any station. There are no arcs between different arrival-departure tracks. Track uniqueness will be satisfied with the flow balance constraints.

2. Station main lines - nonstop trains in the revised timetable cannot dwell on the station main lines. No waiting arcs are constructed from virtual stop points in the space-time railway network.

3. Maintenance constraints - arcs that conflict with maintenance intervals are removed from the space-time network in a pre-processing step.

4. Solution methods

In this section we describe two methods for solving the formulation proposed in Section 3. In Section 4.1 we present an LR approach, while in Section 4.2 we describe an ADMM based method. Both techniques induce train specific subproblems that we solve using dynamic programming. The dynamic programming approach is described in Section 4.3.

4.1. Lagrangian Relaxation

LR is a well-known decomposition method to solve general integer programming problems and has been applied in areas such as routing and scheduling (Fisher, 1981). This method dualizes so-called coupling constraints. This involves removing a subset of constraints from the formulation. Violation of the removed constraints is penalized in the objective function through Lagrange multipliers. Coupling constraints typically link large sets of variables. Removing them allows the problem to be broken into several smaller, independent subproblems, which can be solved more efficiently. LR approaches have been widely used to solve the TTP, see e.g., Caprara et al. (2006) and Brännlund et al. (1998). When using this approach one typically removes from the formulation headway constraints and track capacity constraints, since these couple sets of trains, moves them to the objective function and penalizes any violation. The resulting formulation then decomposes into independent train specific subproblems. When applying LR to the problem at hand we introduce five sets of non-negative Lagrange multipliers \( \lambda_{1}^{i,j,t,r} \), \( \lambda_{2}^{i,j,t,r} \), \( \lambda_{3}^{i,j,t,r} \), \( \lambda_{4}^{i,j,t,r} \), and \( \lambda_{5}^{i,j,t,r} \). These are associated with constraints (12), (13), (14), (15), and (17), respectively. For convenience we again rewrite...
Simultaneously re-optimizing timetables and platform schedules

\[ \lambda_{1}^{i,j,t} \text{ as } \lambda_{1}^{g} \forall g \in A_{se}, \lambda_{4}^{i,j,t} \text{ as } \lambda_{4}^{g} \forall g \in A_{a}, \text{ and } \lambda_{5}^{i,j,t} \text{ as } \lambda_{5}^{g} \forall g \in A_{st}. \]  

The full LR model, which we refer to as model STNM-LR, is given below:

\[
\min Z_2 = \sum_{f \in F} C \cdot \theta_f + \sum_{f \in F} \sum_{g \in A_f} x_{g}^{f} \cdot c_{g}^{f} + \sum_{g \in A_{se}} \lambda_{1}^{g} \cdot R_{1}^{g} + \sum_{k} \sum_{(i, t)} \lambda_{2}^{i,j,k} \cdot R_{2}^{i,j,k} \\
+ \sum_{k} \sum_{(i, t)} \lambda_{3}^{i,j,k} \cdot R_{3}^{i,j,k} + \sum_{g \in A_{a}} \lambda_{4}^{g} \cdot R_{4}^{g} + \sum_{g \in A_{st}} \lambda_{5}^{g} \cdot R_{5}^{g}  
\]

(23)

where

\[
R_{1}^{g} = \sum_{(i', j', t', t') \in \psi_{g}^{f}} x_{i', j', t', t'}^{f} - 1
\]

\[
R_{2}^{i,j,k} = \sum_{(i', j', t', t') \in \psi_{i,j,t}^{f}} x_{i', j', t', t'}^{f} - 1
\]

\[
R_{3}^{i,j,k} = \sum_{(i', j', t', t') \in \psi_{i,j,t}^{f}} x_{i', j', t', t'}^{f} - 1
\]

\[
R_{4}^{g} = \sum_{(i', j', t', t') \in \psi_{g}^{f}} (x_{i', j', t', t'}^{f} + y_{g}^{f}) - 1
\]

subject to constraints (6)–(11), (16), and (18)–(22).

The objective function (23) can be rewritten as (24), where \( \lambda_{n} \) denotes the multipliers associated with the \( n \)th set of relaxed constraints. Note that \( \psi_{g}^{f} \) is new notation here and indicates the arcs whose incompatible arc set contains arc \( g \). As can be seen from the formulation, Model STNM-LR can be divided into independent, train-specific subproblems, (Mahmoudi and Zhou, 2016). For a given set of Lagrange multiplier values, each of these subproblems can be modelled as a resource constrained shortest path problem, which can be efficiently solved using dynamic programming. The Lagrange multipliers affect the arc costs in the space-time network for a train and can be interpreted as an extra penalty for occupying the arc.

The objective function of Model STNM-LR minimizes the original cost plus any penalties incurred from violating the relaxed constraints. If a constraint is violated, a positive Lagrange multiplier increases the cost of occupying the corresponding resource and attempts to reduce the level of occupation. The greater the magnitude of violation for a specific resource, the higher the cost of the associated arc. If any of the relaxed constraints is not violated at the current iteration, its Lagrange multiplier will be reduced to a value not less than zero, or will keep the current value.

\[
\min Z_3 = \sum_{f} \left[ Z_{LR}^{f} \right] - \sum_{n} \lambda_{n}  
\]

(24)

where

\[
Z_{LR}^{f} = \sum_{g \in A_{f}} x_{g}^{f} \cdot c_{g}^{f} + \sum_{g \in A_{f}} \left[ \sum_{(i', j', t', t') \in \psi_{g}^{f}} x_{i', j', t', t'}^{f} \cdot \lambda_{1}^{i', j', t', t'} \right] + \sum_{g \in A_{f}} \left[ \sum_{t' \in (\hat{o}_{g} - T_{g} + T_{r}, \hat{d}_{g})} x_{g}^{f} \cdot \lambda_{2}^{o_{g}, d_{g}, t' \cdot k(\hat{o}_{g})} \right] \\
+ \sum_{g \in A_{f}} \left[ \sum_{t' \in (\hat{d}_{g} - T_{g} + T_{r}, \hat{d}_{g})} x_{g}^{f} \cdot \lambda_{3}^{d_{g}, t' \cdot k(\hat{d}_{g})} \right] + \sum_{g \in A_{f}} \left[ \sum_{(o_{g}, d_{g}, t', t') \in \psi_{g}^{f}} x_{g}^{f} \cdot \lambda_{4}^{o_{g}, d_{g}, t', t'} \right] \\
+ \sum_{g \in A_{f}} \left[ \sum_{(i', j', t', t') \in \psi_{g}^{f}} x_{i', j', t', t'}^{f} \cdot \lambda_{5}^{i', j', t', t'} \right] + \sum_{g \in A_{f}} y_{g}^{f} \cdot \lambda_{5}^{g} + \sum_{g \in A_{f}} y_{g}^{f} \cdot \lambda_{5}^{g} + C \cdot \theta_f
\]
Simultaneously re-optimizing timetables and platform schedules

Figure 15: The symmetry issues with the LR method.

\[ z = \sum_{g \in A} x_g' \cdot \hat{c}_g + C \cdot \theta_f \]

As is the case with most conventional LR approaches, see e.g. Meng and Zhou (2014); Zhou and Teng (2016) and Caprara et al. (2002), we use a subgradient method to iteratively update the Lagrange multipliers. Assuming \( \lambda_n(m) \) is the value of the \( n \)th set of Lagrange multiplier at iteration \( m \), we compute \( \lambda_n(m + 1) \) as follows:

\[
\lambda_n(m + 1) = \max \{ 0, \lambda_n(m) + \alpha_n(m) \cdot R_n(m) \} 
\]

\( \alpha_n(m) = 1/(m + 1) \) is the step size which is not changed for a certain number of iterations, i.e., 10. For a more detailed explanation of LR we refer the reader to Mahmoudi and Zhou (2016), Meng and Zhou (2014), and Zhou and Teng (2016).

4.2. ADMM-based Approach

The proposed formulation can suffer from symmetry issues, and LR struggles to deal with this when it sets train independent Lagrange penalties (Niu et al., 2018). As shown in Figure 15, the original path of train 1 is similar to train 2 just shifted forward in time. Due to scheduled maintenance, trains 1 and 2 have to be delayed and are in this case identical. The two trains will choose between paths 1 and 2. Both trains see the same costs for the two paths and thus will be assigned to the same path in every iteration. An ADMM-based algorithm further augments the LR objective function (24) with additional, quadratic penalty terms and integrates a cyclic block coordinate descent method to solve the augmented Lagrangian. The difference between LR and ADMM is that train subproblems are solved independently with LR, while this is not the case with ADMM. In the ADMM approach the train subproblems are solved sequentially, with the results from previously solved subproblems fixed at each iteration. By introducing train specific task dependent penalties in the ADMM-based algorithm, the symmetry issues can be removed to some extent. As a result, in Figure 15, train 1 will be first assigned to path 1. The cost increase of path 1 seen by train 2 is the quadratic penalty incurred as a result of assigning path 1 to train 1. This ensures that both trains see different costs for each path in any iteration. ADMM-based algorithms have previously been applied with success to time discretized network problems, see e.g, Yao et al. (2019); Zhang et al. (2019d); Tong et al. (2019); Chen et al. (2020).

To apply ADMM, we modify Model STNM-LR by augmenting (24) with additional, quadratic multipliers to penalize any violation of the relaxed set of constraints. Equality constraints can be directly relaxed because any violation should be punished (Yao et al., 2019). However, when we apply the quadratic penalty to less-than-or-equal-to constraints, i.e., (17), if no violation occurs, then no quadratic penalty should be enforced. In order to transform this type of constraint to an equality constraint, a set of continuous, non-negative slack variables \( s_{i,j,k}, \forall k \in K, \forall (i, t) \in N_t^d \cup N_t^d \) and \( s_g, \forall g \in A^a \cup A^e \cup A^{st} \) are added to each coupling constraint (Zhang et al., 2019d). The proposed ADMM model, (STNM-ADMM), can then be written as follows:

\[
\min Z_4 = (24) + \frac{\rho_1}{2} \sum_{g \in A} [R^g_1 + s_g]^2 + \frac{\rho_2}{2} \sum_{k} \sum_{(i,j) \in N_t^d} [R^i,t,k + s_{i,f,k}]^2 \\
+ \frac{\rho_3}{2} \sum_{k} \sum_{(j,r) \in N_t^d} [R^j,t,k + s_{j,t,k}]^2 + \frac{\rho_4}{2} \sum_{g \in A^a} [R^g_4 + s_g]^2 + \frac{\rho_5}{2} \sum_{g \in A^e} [R^g_5 + s_g]^2
\]
Simultaneously re-optimizing timetables and platform schedules

subject to Constraints (6) − (11), (16), (18)−(22) and (27)−(28). The slack variable domains are given below:

\[ 0 \leq s_{i,t,k} \leq 1 \quad \forall k, \forall (i, t) \mid i \in N^d_k \cup N^d_k \tag{27} \]
\[ 0 \leq s_g \leq 1 \quad \forall g \in A^d \cup A^{se} \cup A^{st} \tag{28} \]

The Lagrange multipliers are updated in much the same way as the LR method, as shown in (29). However, the step size here is \( \rho \) rather than \( \alpha \) (Zhang et al., 2019d; Yao et al., 2019). The value of \( \rho \) is treated differently to \( \alpha \). In Zhang et al. (2019d), the value of \( \rho \) is increased according to the violation of the constraints. In our work, however, we treat this as a constant.

\[
\lambda_n(m + 1) = \max \{ 0, \lambda_n(m) + \rho_n \cdot R_n(m) \} \tag{29}
\]

For the sake of clarity, we let \( F' = F \setminus \{ f \} \) denote the train set except the current considered train \( f \). The train specific subproblem objective function can be linearized and is given below. A detailed proof of this is provided in Appendix A.

\[
\min Z_f^{ADM} = \sum_{g \in A_f} x^f_g \cdot c^f_g + \sum_{g \in A^d_f} x^f_g \cdot D^g_1 + \sum_{k \in s^2 \cap N_g} \sum_{g \in N^d_g} x^f_g \cdot D^g_2 + \sum_{k \in s^3 \cap A^d_g} \sum_{g \in N^d_g} x^f_g \cdot D^g_3 + \sum_{g \in A^d_f} \sum_{g \in A^d_f} y^f_g \cdot D^g_4 + \sum_{g \in A^d_f} \sum_{g \in A^d_f} y^f_g \cdot D^g_5 + C \cdot \theta_f
\]

where

\[
D^g_1 = \sum_{(i', j', t', \tilde{r}') \in \Psi^5_g} \left\{ \lambda^f_{i', j', t', \tilde{r}'}, k + \max 0, \frac{\rho_1}{2} \left[ 2 \sum_{(i'', j'', t'', \tilde{r}'')} x^f_{i'', j'', t'', \tilde{r}''} \sum_{f' \in F'(i', j', t', \tilde{r}'')} x^f_{i', j', t', \tilde{r}'} - 1 \right] \right\}
\]
\[
D^g_2 = \sum_{t' \in (d_t - T^k - T^a)} \left\{ \lambda^f_{t', k} + \max 0, \frac{\rho_2}{2} \left[ 2 \sum_{t'' \in (d_t - T^k - T^a)} \sum_{g \in A^d_f} \sum_{f' \in F'(t', j', t', \tilde{r}'')} x^f_{t', j', t', \tilde{r}'} - 1 \right] \right\}
\]
\[
D^g_3 = \sum_{t' \in (d_t - T^k - T^a)} \left\{ \lambda^f_{t', k} + \max 0, \frac{\rho_3}{2} \left[ 2 \sum_{t'' \in (d_t - T^k - T^a)} \sum_{g \in A^d_f} \sum_{f' \in F'(t', j', t', \tilde{r}'')} x^f_{t', j', t', \tilde{r}'} - 1 \right] \right\}
\]
\[
D^g_4 = \sum_{(o_g, d_g, t', \tilde{r}') \in \Psi^5_g} \left\{ \lambda^f_{o_g, d_g, t', \tilde{r}'}, k + \max 0, \frac{\rho_4}{2} \left[ 2 \sum_{(o_g, d_g, t', \tilde{r}') \in \Psi^5_g} \sum_{f' \in F'(o_g, d_g, t', \tilde{r}'')} x^f_{o_g, d_g, t', \tilde{r}'} - 1 \right] \right\}
\]
\[
D^g_5 = \sum_{(i', j', t', \tilde{r}') \in \Psi^5_g} \left\{ \lambda^f_{i', j', t', \tilde{r}'}, k + \max 0, \frac{\rho_5}{2} \left[ 2 \sum_{(i'', j'', t'', \tilde{r}'')} x^f_{i'', j'', t'', \tilde{r}''} + \sum_{f' \in F'(i', j', t', \tilde{r}'')} y^f_{i', j', t', \tilde{r}'} - 1 \right] \right\}
\]
\[
D^g_6 = \sum_{(i', j', t', \tilde{r}') \in \Psi^5_g} \left\{ \lambda^f_{i', j', t', \tilde{r}'}, k + \max 0, \frac{\rho_6}{2} \left[ 2 \sum_{(i'', j'', t'', \tilde{r}'')} x^f_{i'', j'', t'', \tilde{r}''} + \sum_{f' \in F'(i', j', t', \tilde{r}'')} y^f_{i', j', t', \tilde{r}'} - 1 \right] \right\}
\]
Simultaneously re-optimizing timetables and platform schedules

The ADMM-based approach is based on dualization, augmentation, decomposition, and linearization. A more detailed overview of the method is given in Algorithm 1. If no feasible solution is found for a certain train $f$ at a given iteration, a cancellation cost will be directly added to the upper bound calculation. However, train cancellation at one iteration has no influence on later iterations, and all trains will be re-considered at every iteration. We define three termination criteria. The first is the predetermined maximum number of iterations. The second is the maximum time limit. The third uses the optimality gap, which is based on the difference between the upper and lower bounds.

Algorithm 1: ADMM-based approach

**Step 1: Initialization**
Initialize $m \leftarrow 0$, $\lambda_n(m)$, $\rho_n$, best upper bound $UB^* \leftarrow +\infty$ and best lower bound $LB^* \leftarrow -\infty$ as well as corresponding solutions $\{X^*_U\} \text{ and } \{X^*_L\}$

**Step 2: Generate the lower bound solution by solving pure Lagrangian problem**
for $g \in A$ do
Update $\hat{c}_g$ according to Eq. (24)
end for
for $f \in F$ do
Compute train $f$’s shortest path using dynamic programming
end for
Compute $LB^m$ by Eq. (23); update $LB^* = \max \{LB^*, LB^m\}$ and $\{X^*_L\}$

**Step 3: Generate the ADMM solution**
for $f \in F$ do
for $g \in A_f$ do
Update $\hat{c}_f$ according to Eq. (30)
Compute train $f$’s shortest path using dynamic programming
end for
end for

**Step 4: Generate the upper bound solution by Algorithm 2**
if conflicts between trains paths in the ADMM solution exist then
Transform the ADMM solution to feasible solution by calling Algorithm 2.
end if
Compute $UB^m$ by Eq. (5); update $UB^* = \min \{UB^*, UB^m\}$ and $\{X^*_U\}$

**Step 5: Update the Lagrange multipliers**
$m \leftarrow m + 1$; update $\lambda_n(m)$ based on the sub-gradient method by Eq. (29)

**Step 6: Termination condition test**
If the termination condition is satisfied, stop; Otherwise, return to step 2.

Priority-based rules are typically used to produce a feasible upper bound for the LR model. Examples of these include cost-based priority, conflict-number based priority, and train-sequence-based priority, as suggested by Meng and Zhou (2014); Mahmoudi and Zhou (2016) and Zhou and Teng (2016), respectively. In our work, we add randomness to a train ranking procedure. If the train sequence has been calculated at a previous iteration, then two trains will be chosen randomly to generate a new train sequence, increasing the likelihood of improving the incumbent solution. This algorithm is outlined in Algorithm 2.

4.3. Time-dependent dynamic programming algorithm

The application of LR or an ADMM-based method results in a formulation that decouples into individual train subproblems. For a given set of Lagrange multipliers and quadratic penalties, each of these subproblems involves solving a resource constrained shortest path, and for this we propose a dynamic program. Dynamic programming algorithm has been widely applied to solve the resource-constrained shortest path problem, see e.g., Mahmoudi and Zhou (2016); Shang et al. (2019).

A conventional label setting algorithm is used to find the shortest, resource constrained, path for a given train in its space-time network. We monitor two resources, cost and waiting (dwell) time. At any stopping point there is a minimum and maximum dwell time that must be respected. Labels are generated in a forward pass through the space-time network for a given train starting from the origin node. Successive labels are generated through an extension procedure by considering all of a node’s successors. Each label stores information on the resident node, time index, its predecessor label (the label from which it was generated), and the values of the two resources. Extensions that violate the dwell time restrictions are not considered. The shortest path can be easily found by determining the “best” labels at the destination node and then tracing movements of the train back through each of the
Algorithm 2 Heuristic method to generate the feasible upper bound solution

Step 1: Train ranking
Rank \( f \in F \) by the increasing number of conflicting trains.
If the train sequence has repeated more than (include) once, choose two trains randomly to generate a new train sequence.

Step 2: Schedule trains sequentially and generate a feasible solution
Set the available arcs for all trains to be \( A_{\text{available}} \)
Cancelled train set \( F_c \leftarrow \emptyset \)
for \( f \in F \) (according to the ranking in Step1) do
if train \( f \) without conflicting trains then
    Apply the path in the current ADMM solution directly
else
    Find \( f \)'s shortest path from \( A_{\text{available}} \) using dynamic programming
    if no feasible solution for train \( f \) then
        \( F_c = F_c \cup f \)
    end if
end if
Remove the arcs that have been chosen by train \( f \) from \( A_{\text{available}} \)
end for

A dominance procedure is applied to reduce the total number of labels considered. Dominance involves comparing two labels for a given vertex and tries to determine whether one of the two labels can be discarded. For receiving and departure nodes, the dominance test involves comparing only the levels of the cost resource. Assuming we have two labels resident at the same waiting node and that they have respective costs \( c_1 \) and \( c_2 \), the first label dominates the second if \( c_1 < c_2 \), while the second dominates the first if \( c_2 < c_1 \). If \( c_1 = c_2 \), only one label is removed. For waiting nodes the dominance test is somewhat more complicated since it involves both resource levels. Let us assume that we have two labels resident at the same waiting node and that they have respective costs \( c_1 \) and \( c_2 \) and respective waiting times of \( w_1 \) and \( w_2 \). If \( c_2 < c_1, w_2 \geq w_1 \) or \( c_2 = c_1, w_2 > w_1 \), the second label dominates the first. However, if \( c_2 > c_1, w_2 > w_1 \), then we cannot say that the first label is always better than the second, and we therefore keep both. Due to the minimum dwell time restriction, it maybe possible that the second label is better than extensions of the first label. In this case, dominance is only possible as long as \( w_1 \) and \( w_2 \) both satisfy the minimum dwell time constraint.

5. Computational results

In this section we test and compare the performance of LR and the ADMM-based approach for several instances based on two different networks. The first network is a small virtual network, while the other is an actual network from the Chinese high-speed railway and is significantly larger. For the small network, it is possible to compute optimal solutions using the commercial solver CPLEX 12.9, and these results are added to the comparison. The LR approach and the ADMM-based approach are implemented in Python 3.7. All instances are tested on a personal Windows computer with a 2.50GHz processor and 8GB of memory.

5.1. A small virtual network

The small virtual network is shown in Figure 16. It is comprised of three stations, containing 24 nodes and 40 physical arcs. All track sections and main lines within stations are one-way and can be chosen by trains from the corresponding direction. Station platform tracks are, however, bidirectional and can be occupied by both outbound and inbound trains. The running time on receiving and departure arcs is set to one minute, while the travel time on the section arcs is assumed to be five minutes. All outbound trains originate at Node 1 and leave the network at Node 23, while all inbound trains originate at Node 24 and leave the network at Node 2.
Simultaneously re-optimizing timetables and platform schedules

Figure 16: Simple virtual railway network.

The dwell time of all trains at each station in the original timetable is set to zero, one, or two minutes. A dwell time of zero minutes indicates a nonstop train. There is no shift, stretch, or conflicts, in the artificial original timetable.

The minimum dwell time is set to an ideal dwell time. All trains cannot dwell at a certain station for more than 10 minutes. The allowed time window in which one can shift the departure of a train at its origin node is set to \([0 \text{ minutes}, 20 \text{ minutes}]\) for all trains, and the maximum delay time at the destination node is assumed to be 20 minutes. The values of all types of Lagrange multipliers belong to the range \([1,4]\). The values of the other required parameters that are used in the small case are listed in Table 8. For this particular problem we assumed station track maintenance is scheduled for the time interval \([15,40] (0 \text{ refers to the start of the planning horizon}) at Node 14 and on the corresponding physical arcs \{(9,14), (14,10), (14,15), (16,14)\}.

Table 8

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T_{d}^{i,j})</td>
<td>3 min</td>
</tr>
<tr>
<td>(T_{a}^{i,j})</td>
<td>3 min</td>
</tr>
<tr>
<td>(T_{s}^{i})</td>
<td>1 min</td>
</tr>
<tr>
<td>(T_{r}^{i})</td>
<td>1 min</td>
</tr>
<tr>
<td>(T)</td>
<td>80 min</td>
</tr>
<tr>
<td>(C)</td>
<td>(10 \times T)</td>
</tr>
</tbody>
</table>

We investigate seven instances for this network. Each instance has between 4 and 16 trains. Table 9 reports the performance of CPLEX, LR, and the ADMM-based approach. For CPLEX we report the optimal objective value \((Z^{*})\) and the time taken \((CT)\) in seconds to solve each instance. For the other two methods, we report the value of upper and lower bounds, the time taken, the number of total iterations \((TI)\), the iteration number at which the best upper bound was found \((IB)\), the percentage gap between the upper bound and the optimal objective value \((GAP^{1})\), and the percentage gap between upper bound and lower bound obtained \((GAP^{2})\). As can be seen from Table 9, both LR and ADMM produce the optimal solution for the instances with four and six trains solution as CPLEX solver in the cases with a small number of trains, but take longer. For cases with more trains, CPLEX performs better but takes considerably longer to prove optimality. The optimality gaps for LR and ADMM increase as the problem size grows and both methods struggle to find a good lower bound. The upper bounds provided by ADMM are at least as good, and in many cases better, than the upper bounds provided by LR. Furthermore, ADMM achieves the bound in fewer iterations. Each iteration is, however, computationally heavier. With the exception of the instances with 10 trains and 14 trains, LR does, however, provide tighter lower bounds. For large-scale instances, CPLEX is unable to provide any feasible solution. We are interested in finding the best upper bound, and from this perspective, ADMM is better.

To visually show the benefit of a flexible station track utilization policy compared to a fixed rule, Figure 17 and Figure 18 illustrate the optimal solution for each policy for the instance with 16 trains. Two trains are cancelled if a fixed policy is applied, while all trains can be scheduled when the flexible policy is used. In Figures 17 and 18, the red line indicates the station arrival-departure track that is undergoing maintenance and the purple lines indicate the use of a reverse arrival-departure track. As shown in Figure 17, not only do trains use reverse tracks in Station 2, where the maintenance is, but also in Station 3. This allows trains to
Simultaneously re-optimizing timetables and platform schedules

Table 9
Performance comparison between LR-based and ADMM-based methods as well as CPLEX solver on the small network.

| $|F|$ | $Z^*$ | CPU(s) | UB  | LB  | CT(s) | TIs | IB  | GAP$^1$ | GAP$^2$ |
|-----|------|-------|------|-----|-------|-----|-----|---------|---------|
| 4   | 102.0| 0.4   | 102.0| 100.5| 19.6  | 10  | 7   | 0.0%    | 1.5%    |
| 6   | 184.0| 3.5   | 184.0| 159.6| 28.6  | 10  | 1   | 0.0%    | 13.3%   |
| 8   | 296.0| 49.8  | 298.0| 224.7| 39.7  | 10  | 1   | 0.7%    | 24.6%   |
| 10  | 424.0| 466.8 | 465.0| 288.0| 102.6 | 20  | 15  | 9.7%    | 38.1%   |
| 12  | 569.0| 1230.7| 595.0| 351.0| 122.0 | 20  | 16  | 4.6%    | 41.0%   |
| 14  | 639.0| 2972.3| 719.0| 389.7| 145.0 | 20  | 11  | 12.5%   | 45.8%   |
| 16  | 721.0| 3585.0| 803.0| 441.3| 255.8 | 30  | 3   | 11.4%   | 45.0%   |

Note: $Z^*$ refers to the optimal objective value; UB and LB indicate the upper bound and lower bound; CT means CPU time; TIs denotes total iterations calculated in current case; IB means iteration number in which obtain the best upper bound; GAP$^1$ = $|UB - OS|/OS$ and represents the gap between optimal solution and best upper bound; GAP$^2$ = $|UB - LB|/UB$ and denotes the gap between upper bound and lower bound.

Figure 17: Optimal result for 16 trains with a flexible track policy.

utilize the track resources more effectively.

As an additional experiment, we compare the performance of the fixed track utilization policy with a flexible one on four different instances for the case with 16 trains. The four cases differ in the duration of the scheduled maintenance at Node 14. Table 10 summarizes the results obtained with the ADMM approach. All parameters are set to the values specified in Table 8. Node 14 is the platform track associated with inbound trains. Under a fixed track utilization policy, inbound trains are solved independently and outbound trains are not adjusted. Both inbound and outbound trains simultaneously adjusted when using the flexible utilization policy.

Under a flexible track utilization policy, all trains can be scheduled when adjusting the timetable for the first two instances. However, under a fixed track utilization policy, the station track maintenance at Station B impacts inbound stopping trains and leads to several cancellations. Furthermore, longer maintenance windows lead to a greater number of cancellations. Two inbound trains are cancelled when the duration of maintenance is 25 minutes in two cases. The results for the last two cases are the same when the duration of maintenance is 50 minutes or 70 minutes. The reason for this is that, in each case, the maintenance duration at Node 14 is considerably longer than the occupation time of trains in the original timetable. Only three trains out of sixteen cannot be timetabled under a flexible track utilization policy, while six out of eight trains are cancelled under a fixed rule. Two trains that are not cancelled under the fixed policy are nonstop trains at the Station 2. Cancelling train service is the last resort for train operating companies and leads to passenger dissatisfaction. Therefore, a flexible station track utilization policy might be preferable as it can improve the quality of the revised timetable. Simultaneously optimizing trains both directions, does, however, take slightly longer than the separately solving each direction in every iteration.
Simultaneously re-optimizing timetables and platform schedules

![Diagram showing timetables and platform schedules with a red line indicating two inbound trains being cancelled.]

**Figure 18:** Optimal result for 16 trains with a fixed track policy.

<table>
<thead>
<tr>
<th>Table 10</th>
<th>Comparison of two track utilization policies on the small network (16 trains, ADMM).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Policy</td>
</tr>
<tr>
<td>1</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>Flexible</td>
</tr>
<tr>
<td>2</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>Flexible</td>
</tr>
<tr>
<td>3</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>Flexible</td>
</tr>
<tr>
<td>4</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>Flexible</td>
</tr>
</tbody>
</table>

5.2. A large practical network

We now investigate the performance of LR and ADMM on an actual part of the Chinese high-speed railway network. In particular, we focus on the Wuhan-Guangzhou high-speed railway corridor. An overview of this network is given in Figure 19. This network is comprised of nine stations. In reality, the considered network is part of a much bigger network, and other high-speed lines do cross this corridor (termed cross-lines). We have therefore omitted station track that is dedicated to any high-speed lines that cross the considered network. Trains that run on the cross lines are modified in such a way that they become part of the considered line. This is done in order to capture cross-line interactions (since high-speed trains from different lines share station platforms). This modification step involves updating the first arrival boundary and/or last departure boundary for any cross-line train on the considered network.

With the exception of the length of the planning horizon, which is set to 380 minutes, all parameter values are set to the values given in Table 8. We consider 40 trains in this experiment (20 in each direction). All outbound trains depart from Node 1 or Node 9 and end at Node 96, while inbound trains start from Node 97 and leave the network at Node 2 or Node 10. For a problem of this size, the formulation given in Section 3 cannot be directly solved with CPLEX. We therefore only compare the performance of LR and ADMM for this network. Three scenarios, with different maintenance locations and durations, are designed to compare the performance of the two different solution methodologies. The first scenario considers scheduled maintenance at Station 2 in the window [100, 250], where limits on the time window refer to minutes after the start of the planning horizon. Scenario two considers inbound track maintenance at Station 4 for the duration of the time interval [100, 250]. Finally, the third scenario considers outbound track maintenance at Station 2 for the duration of the time interval [140, 215]. The maximum shift at the origin node and the maximum delay at the destination node are assumed to be 50 minutes. These values are set by a dispatcher in practice and...
Simultaneously re-optimizing timetables and platform schedules

depend on a trade-off between train delay and cancellation. The maximum number of iterations each algorithm can perform is 30.

Tables 11, 12 and 13 present the computational results for each of three scenarios. For the ADMM-based method, we report results for different sets of values for the penalties $\rho_1$, $\rho_2$, $\rho_3$, $\rho_4$, and $\rho_5$. The tables report the upper and lower bounds, along with the optimality gaps. ADMM finds better upper bounds than LR, irrespective of the penalty value setting. Furthermore, the optimality gaps are also better. On the large problem instances there is not such a large difference between the lower bound values for the two approaches. The results indicate that the proposed ADMM-based algorithm outperforms LR on the larger network. The optimality gaps for the larger instances are noticeably smaller than those of the small instances (particularly when the maintenance tasks are at Station 2). In the small network, almost all trains can only use track 11 when track 14 undergoes maintenance. In the lower bound calculation, the trains prefer to use track 11 rather than incur the waiting cost at the origin node. Observe that in a lower bound calculation trains are – allowed – to conflict with each other. The waiting cost at the origin node accounts for a large proportion of the total cost, as trains must wait at the origin node in any feasible solution. Therefore the optimality gap is influenced by the limited track possibilities and the number of trains (the gap increases as the number of trains increases). A similar observation can be made for the large network, although the effect is not as pronounced. Larger optimality gaps are obtained when the maintenance task is located at a station with fewer track possibilities. The optimality gaps in Table 12 are almost 8%, depending on the parameter choice. There are many more track possibilities at Station 2, and Tables 11 and 13 show that the optimality gaps are smaller. It is also worth mentioning that the small artificial instance is much more congested (has more trains per unit time) than the large case. This leads to a greater number of potential conflicts. Tables 11 and 13 provide a comparison of the two methods for different maintenance window durations over a fixed planning horizon. The performance also seems to deteriorate as the maintenance window increases. As can be seen from the results, the performance of the ADMM-based approach, in terms of the quality of the upper bound, depends on the different quadratic penalties applied. For each scenario, we show in bold the best optimality gaps, the best upper and lower bounds found by the ADMM-based method with seven given sets of penalties. The best combination of the quadratic penalties clearly depends on the duration and location of the maintenance. In all scenarios the best upper bound can be obtained when the values for the five penalties are six, while the best lower bounds are obtained when the values are equal to one. The results do indicate that the best optimality gaps are found for different penalty value combinations when the track maintenance is scheduled in different locations. It seems that the larger the station size is, the smaller the quadratic penalties must be to obtain the best gap.

**Table 11**

<table>
<thead>
<tr>
<th>Items</th>
<th>LR</th>
<th>ADMM with different combinations of $\rho_1$, $\rho_2$, $\rho_3$, $\rho_4$, $\rho_5$ with</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1,1,1,1,1, 2,2,2,2, 3,3,3,3, 4,4,4,4,4, 5,5,5,5,5, 6,6,6,6,6, 7,7,7,7,7, 8,8,8,8,8, 9,9,9,9,9, 10,10,10,10,10, 11,11,11,11,11, 12,12,12,12,12, 13,13,13,13,13, 14,14,14,14,14, 15,15,15,15,15, 16,16,16,16,16, 17,17,17,17,17, 18,18,18,18,18, 19,19,19,19,19, 20,20,20,20,20, 21,21,21,21,21, 22,22,22,22,22, 23,23,23,23,23, 24,24,24,24,24, 25,25,25,25,25, 26,26,26,26,26, 27,27,27,27,27, 28,28,28,28,28, 29,29,29,29,29, 30,30,30,30,30, 31,31,31,31,31, 32,32,32,32,32, 33,33,33,33,33, 34,34,34,34,34, 35,35,35,35,35, 36,36,36,36,36, 37,37,37,37,37, 38,38,38,38,38, 39,39,39,39,39, 40,40,40,40,40, 41,41,41,41,41, 42,42,42,42,42, 43,43,43,43,43, 44,44,44,44,44,</td>
</tr>
<tr>
<td>Upper bound</td>
<td>5400</td>
<td>5345</td>
</tr>
<tr>
<td>Lower bound</td>
<td>5320</td>
<td>5315</td>
</tr>
<tr>
<td>GAP$^2$</td>
<td>1.48%</td>
<td><strong>0.56%</strong></td>
</tr>
</tbody>
</table>

Figure 19: Illustration of the practical network.
Simultaneously re-optimizing timetables and platform schedules

Table 12
Results of maintenance at all inbound tracks in Station 4 from 100min to 250min.

<table>
<thead>
<tr>
<th>Items</th>
<th>LR</th>
<th>ADMM with different combinations of $\rho_1$, $\rho_2$, $\rho_3$, $\rho_4$, $\rho_5$ with</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1,1,1,1,1</td>
</tr>
<tr>
<td>Upper bound</td>
<td>5964</td>
<td>5714</td>
</tr>
<tr>
<td>Lower bound</td>
<td>5294</td>
<td>5287</td>
</tr>
<tr>
<td>GAP$^2$</td>
<td>11.24%</td>
<td>7.47%</td>
</tr>
</tbody>
</table>

Table 13
Results of maintenance at all inbound tracks in Station 2 from 140min to 215min.

<table>
<thead>
<tr>
<th>Items</th>
<th>LR</th>
<th>ADMM with different combinations of $\rho_1$, $\rho_2$, $\rho_3$, $\rho_4$, $\rho_5$ with</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1,1,1,1,1</td>
</tr>
<tr>
<td>Upper bound</td>
<td>5307</td>
<td>5281</td>
</tr>
<tr>
<td>Lower bound</td>
<td>5272</td>
<td>5269</td>
</tr>
<tr>
<td>GAP$^2$</td>
<td>0.67%</td>
<td>0.23%</td>
</tr>
</tbody>
</table>

The lower bounds provided by LR are, in all cases, tighter than those which are provided by the ADMM-based method. This was also observed by Yao et al. (2019), who pointed out that the ADMM can usually find upper bounds of high quality, but struggles with finding good lower bounds. Other methods should be applied to get a lower bound to estimate the corresponding solution quality. Finding better lower bounds for the problem at hand is a challenging problem and the subject of future research.

As we did for the small network in Section 5.1, we perform several experiments to analyse the impact of the flexible track utilization policy for the larger network. Here, we define five scenarios. The first two scenarios focus on station track maintenance, while the last three scenarios investigate the track segment maintenance. Scenarios 1 and 2 are focus on the inbound tracks at Stations 2 and Station 4, respectively. The time window for the scheduled maintenance is [100, 250]. We do not enforce any time windows on the time shift allowed origin and destination nodes. This is in order to obtain a revised timetable without any cancellations under the fixed track utilization policy. Scenario 3 is that track segments in both directions between Stations 3 and 4 and Stations 4 and 5 are scheduled to be blocked for duration of the time window [150, 190]. Scenario 4 is similar to Scenario 3 except that only inbound tracks are blocked. All track segments between Stations 4 and 5 and Stations 5 and 6 are scheduled to be blocked for duration of the window [150, 190] in Scenario 5. For Scenarios 3, 4 and 5, we permit a time shift anywhere on the interval [0, 60] at a train’s origin node and allow a maximum delay of 90 minutes for each train, enforced at the destination nodes. To fully investigate the track utilization, we set $a_1 = 6$. The two different station track utilization policies are implemented for every scenario.

Table 14 provides a summary of the results. For each maintenance scenario under each track utilization policy, we report the number of trains using reverse station track and the number of trains rerouted at each station. We also report the total amount of delay (in minutes). Under the fixed track utilization policy, the track segment maintenance blocks tracks in both directions in Scenarios 3 and 5, and the two directions are optimized separately. Therefore, we report the results for the two different directions. The Total train delay under the flexible track utilization rule is less than that of the fixed policy for the first four scenarios. This confirms that the simultaneous optimization of trains from both directions results in a revised timetable of better quality than separately optimizing each direction. Although the total delay under the fixed track utilization policy is less than that for the flexible policy in Scenario 5, one train is cancelled in the fixed case. When a flexible track policy is adopted, reverse tracks provide the possibility for station capacity to be shared by trains from two directions. This reduces the total delay time. It is also evident from the results that a flexible track utilization policy reduces the total delay time much more for station track maintenance than it does for track segment maintenance.

With the flexible track utilization rule, trains only occupy the reverse station track in the stations that are impacted by the maintenance, such as Station 4 and Station 2 in the first two scenarios. This is because trains can take full advantage of reverse tracks during the maintenance window to recover the traffic rapidly. However, for track segment maintenance, reverse station tracks are used at a wider range of stations. The same observation is also true when it comes to rerouting. If station track maintenance is performed, then rerouting tends to happen at that specific station. Rerouting occurs at a wider set of stations when track segment maintenance is performed.

Under the fixed track utilization policy no trains are rerouted at Station 4 in the first scenario. Because only one platform track in each direction can be utilized to reduce the delay. Three trains are rerouted at Station 2 in the second scenario. The reason for this is that there are multiple platform tracks in each direction and therefore rerouting can reduce the total delay. Rerouting obviously has greater potential at larger stations due to more routing possibilities. Most rerouting occurs at the stations prior to the station impacted by the station track maintenance with this fixed rule.

Qin Zhang et al.: Page 29 of 34


Simultaneously re-optimizing timetables and platform schedules

**Table 14**
Comparison of several maintenance scenarios on the practical network.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Rule</th>
<th>Reverse</th>
<th>Rerouting</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>S1 S2 S3 S4 S5 S6 S7 S8 S9</td>
<td>S1 S2 S3 S4 S5 S6 S7 S8 S9</td>
<td></td>
</tr>
<tr>
<td>1 - S4/IN</td>
<td>Flexible</td>
<td></td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Fixed</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2 - S2/IN</td>
<td>Flexible</td>
<td></td>
<td>19</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Fixed</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>3 - S3, S4 &amp; S4, S5</td>
<td>Flexible/IN</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Flexible/OUT</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Fixed/IN</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Fixed/OUT</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>4 - S3, S4 &amp; S4, S5/IN</td>
<td>Flexible</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Fixed</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>5 - S4, S5 &amp; S5, S6</td>
<td>Flexible/IN</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Flexible/OUT</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Fixed/IN</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Fixed/OUT</td>
<td></td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

Note: S1~S9 denote Station1~Station9; IN represents the inbound running direction; OUT represents the outbound running direction; C is the train cancellation

For the track segment maintenance scenarios, local rerouting occurs at the stations before the blocked sections regardless of the train’s running direction under both track utilization rules. Outbound and inbound trains have to wait at the stations before the blocked sections, making it impossible for trains to use their originally planned route. Therefore, the rerouting strategy is used to adjust the platforming schedules. A flexible track utilization policy provides more possibilities to improve the quality of the adjusted timetable and routes. The results for the last three scenarios indicate that rerouting trains in several stations is more beneficial than re-platforming trains at stations impact by the scheduled maintenance.

6. Conclusion and future research

In this paper, we have proposed two methods, LR and an ADMM-based method, to simultaneously re-optimize a high-speed train timetable and platform allocation under scheduled track maintenance. Furthermore, we have assessed the impact of a flexible track utilization policy versus a conventional fixed track utilization policy. The underlying mathematical formulation models the problem as a binary integer program on a space-time network. The formulation minimizes the number of cancellations and the weighted train travel times. The two decomposition methods result in a series of train specific subproblems that we solve using dynamic programming. Heuristic upper bound procedures are used in conjunction with both procedures to provide optimality gaps on the solution quality. Several instances based on a virtual small-scale network and a practical large-scale railway line are investigated to verify the validity of our proposed model and compare the effectiveness and efficiency of two decomposition methodologies. Computational experiments show that the total delay can be reduced by 95.2%~99.7% if a flexible track utilization policy is applied at the station where track maintenance is scheduled. Furthermore, the number of cancellations for a case with 16 trains decreases by 50% with this policy. According to the results of the cases, we can draw a number of conclusions. These are as follows:

1. The proposed model and algorithms can provide solutions of good quality to the integrated high-speed train timetabling and station routing problem under scheduled maintenance.
2. The ADMM-based method can obtain better upper bounds in fewer iterations than LR, especially for large-scale problems.
3. A flexible track utilization policy that provides the possibility to simultaneously schedule outbound and inbound trains results in better solutions, reducing the number of cancellations and the total delay, compared to a fixed track utilization policy which optimizes trains from different directions independently.

Qin Zhang et al.: Page 30 of 34
Simultaneously re-optimizing timetables and platform schedules

(4) Results on the large scale network show that it is beneficial to reroute trains at multiple stations. This confirms that multiple stations should be considered when adjusting timetables and platform schedules to use the infrastructure as efficiently as possible.

Further research will mainly focus on the following aspects. Firstly, the lower bounds obtained by the ADMM-based approach are quite weak. Therefore, an effective method to obtain a better lower bound should be developed in the future. Secondly, in this paper, we have investigated the TTP and the TPP for a single railway line connecting multiple stations. However, as the Chinese high-speed railway expands, more complicated networks are appearing; single railway lines are crossing more and more lines. This creates more interaction between trains from different lines. The simultaneous optimization of multiple railway lines will result in larger, more complicated problems and is hence an important direction for future research.

Acknowledgements

This research was supported by the National Key R&D Program of China (No.2018YFB1201403).

Appendix A

The objective function of model STNM-ADMM is:

$$\min Z_4 = (24) + \frac{\rho_1}{2} \sum_{g \in A^{et}} [R_{g}^{1} + s_g] + \frac{\rho_2}{2} \sum_{(i,j) \in N^a_g} [R_{i,j,k}^{1} + s_{i,j,k}]^2$$

$$+ \frac{\rho_3}{2} \sum_{k \in N^a} \sum_{(i,j) \in N^q_k} [R_{i,j,k}^{3} + s_{j,r,k}]^2 + \frac{\rho_4}{2} \sum_{g \in A^a} [R_{g}^{4} + s_g] + \frac{\rho_5}{2} \sum_{g \in A^a} [R_{g}^{5} + s_g]$$

Due to the block-by-block update discipline, the model STNM-ADMM can be separated into several subproblems for each train $f$ where the variables associated with other trains are fixed. Here $F'$ represents all trains other than train $f$. All quadratic penalty terms can then be rewritten as the form of:

$$F (x_f) = \left( \sum_{f} x_f + \sum_{f \in F'} x_f + s - 1 \right)^2$$

where $\sum x_f = 0$ or $\sum x_f = 1$. Zhang et al. (2019d) discusses a linearization technique to handles this kind of function and shows how to determine the value of the slack variables. Here we briefly show the linearization process.

$$F (x_f) = \left( \sum_{f} x_f + \sum_{f \in F'} x_f + s - 1 \right)^2$$

$$= (\sum x_f)^2 + 2 (\sum x_f) \left[ \left( \sum_{f \in F'} x_f \right) + s - 1 \right] + \left( \sum_{f \in F'} x_f \right)^2 + s - 1$$

$$= \sum x_f + 2 (\sum x_f) \left[ \left( \sum_{f \in F'} x_f \right) + s - 1 \right] + \left( \sum_{f \in F'} x_f \right)^2 + s - 1$$

$$= \sum x_f \left[ 2 (\sum_{f \in F'} x_f) + 2s - 1 \right] + \left( \sum_{f \in F'} x_f \right)^2 + s - 1$$

To obtain the value of the slack variable, Eq.(33) can be seen as the function with the variable $s$ in Eq.(34).

$$F (s) = s^2 + 2s \left( \sum x_f + \sum_{f \in F'} x_f - 1 \right) + \text{Constant}$$

where $F(s)$ is optimal when $s^* = 1 - \sum x_f - \sum_{f \in F'} x_f$. 

Qin Zhang et al.: Page 31 of 34
Simultaneously re-optimizing timetables and platform schedules

Condition 1: \( \sum \sum_{j \in F^T} x_f = 0 \), which means no other trains occupy considered resources. There should be no penalty from the quadratic term. In this condition, \( s^* = 1 - \sum x_f \) and \( s^* \in [0, 1] \). According to the slack variable domains, \( F^*(s) = F(s^*) = F(1 - \sum x_f) \). Therefore, Eq. (33) can be rewritten as follows:

\[
F(x_f) = \left( \sum x_f + \sum \sum_{j \in F^T} x_f + s - 1 \right)^2 = \left( \sum x_f + 0 + 1 - \sum x_f - 1 \right)^2 = 0
\]

(35)

Condition 2: \( \sum_{j \in F^T} x_f \geq 1 \), which means other trains occupy the considered resources more than once. There should be penalty from the quadratic term. In this condition, \( s^* \leq 0 \). According to the slack variable domains, \( F^*(s) = F(s = 0) \). Therefore, Eq. (33) can be rewritten as follows:

\[
F(x_f) = \sum x_f \left[ 2 \left( \sum_{j \in F^T} x_f \right) - 1 \right] + \left( \sum_{j \in F^T} x_f \right) - 1
\]

(36)

In conclusion,

\[
F(x_f) = \begin{cases} 
0, & \sum_{j \in F^T} x_f = 0 \\
\sum_{j \in F^T} x_f \left[ 2 \left( \sum_{j \in F^T} x_f \right) - 1 \right] + \text{Constant}, & \sum_{j \in F^T} x_f \geq 1
\end{cases}
\]

(37)

Therefore, the quadratic term that should be added to each train subproblem has the form of Eq. (38) without the constant term.

\[
F = \max \left\{ 0, \sum x_f \left[ 2 \left( \sum_{j \in F^T} x_f \right) - 1 \right] \right\}
\]

(38)

References


Qin Zhang et al.:  Page 32 of 34
Simultaneously re-optimizing timetables and platform schedules

Transportation Research Part E: Logistics and Transportation Review 105, 83–104.


Yang, L., Qi, J., Li, S., Gao, Y., 2016. Collaborative optimization for train scheduling and train stop planning on high-speed railways. Omega 64, 57–76.


