Delft University of Technology

Delft Center for Systems and Control

Technical report 18-012

Integration of real-time traffic management and train control for rail networks – Part 1: Optimization problems and solution approaches*

X. Luan, Y. Wang, B. De Schutter, L. Meng, G. Lodewijks, and F. Corman

If you want to cite this report, please use the following reference instead:

X. Luan, Y. Wang, B. De Schutter, L. Meng, G. Lodewijks, and F. Corman, "Integration of real-time traffic management and train control for rail networks – Part 1: Optimization problems and solution approaches," *Transportation Research Part B*, vol. 115, pp. 41–71, Sept. 2018. doi:10.1016/j.trb.2018.06.006

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/18_012.html

Integration of real-time traffic management and train control for rail networks – Part 1: Optimization problems and solution approaches

Xiaojie Luan[a], Yihui Wang[b,*], Bart De Schutter[c], Lingyun Meng[d], Gabriel Lodewijks[e], Francesco Corman[f]

[a]*Section Transport Engineering and Logistics, Delft University of Technology, 2628 CD Delft, the Netherlands*
[b]*State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China*
[c]*Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, the Netherlands*
[d]*School of traffic and transportation, Beijing Jiaotong University, Beijing 100044, China*
[e]*School of Aviation, Faculty of Science, University of New South Wales, Sydney, Australia*
[f]*Institute for Transport Planning and Systems (IVT), ETH Zürich, Stefano-Franscini-Platz 5, 8093 Zürich, Switzerland*

Abstract

We study the integration of real-time traffic management and train control by using mixed-integer nonlinear programming (MINLP) and mixed-integer linear programming (MILP) approaches. Three innovative integrated optimization approaches for real-time traffic management that inherently include train control are developed to deliver both a train dispatching solution (including train routes, orders, departure and arrival times at passing stations) and a train control solution (i.e., train speed trajectories). Train speed is considered variable, and the blocking time of a train on a block section dynamically depends on its real speed. To formulate the integrated problem, we first propose an MINLP problem ($\mathrm{P_{NLP}}$), which is solved by a two-level approach. This MINLP problem is then reformulated by approximating the nonlinear terms with piecewise affine functions, resulting in an MILP problem ($\mathrm{P_{PWA}}$). Moreover, we consider a preprocessing method to generate the possible speed profile options for each train on each block section, one of which is further selected by a proposed MILP problem ($\mathrm{P_{TSPO}}$) with respect to safety, capacity, and speed consistency constraints. This problem is solved by means of a custom-designed two-step approach, in order to speed up the solving procedure. Numerical experiments are conducted using data from the Dutch railway network to comparatively evaluate the effectiveness and efficiency of the three proposed approaches with heterogeneous traffic. According to the experimental results, the MILP approach ($\mathrm{P_{TSPO}}$) yields the best overall performance within the required computation time. The experimental results demonstrate the benefits of the integration, i.e., train delays can be reduced by managing train speed.

Keywords: Real-time traffic management, Train control, Integrated optimization, Delay recovery, Mixed integer linear programming (MILP)

1. Introduction

Railway transport systems are of crucial importance for the competitiveness of national or regional economy as well as for the mobility of people and goods. To improve reliability of train services and increase satisfaction of customers, many railway infrastructure managers (e.g., Network Rail in United Kingdom and Banedanmark in Denmark) and train operating companies (e.g., V/Line in Australia) have set their own targets for train punctuality, in terms of punctuality rates. Moreover, there have been many projects over the years that have aimed at improving the punctuality of trains, such as the On-Time project (Quaglietta et al. 2016). Policy makers and researchers have been seeking approaches for attaining the punctuality goals.

*Corresponding author
Email addresses: x.luan@tudelft.nl (Xiaojie Luan), yihui.wang@bjtu.edu.cn (Yihui Wang), B.DeSchutter@tudelft.nl (Bart De Schutter), lymeng@bjtu.edu.cn (Lingyun Meng), g.lodewijks@unsw.edu.au (Gabriel Lodewijks), francesco.corman@ivt.baug.ethz.ch (Francesco Corman)

In real operations, unavoidable perturbations (caused by bad weather, infrastructure failure, extra passenger flow, etc.) often happen and result in delays to the original train timetable, which make difficulties in achieving the punctuality goals. When trains are delayed from the normal operation, train dispatchers are in charge of adjusting the impacted train timetables from perturbations (by means of taking proper dispatching measures, e.g., re-timing, re-ordering, and re-routing), so as to reduce potential negative consequences (train delays); train drivers are responsible for controlling the delayed trains (by means of taking proper driving actions, i.e., accelerating, cruising, coasting, and braking) to reach the stations at the times specified by train dispatchers, with the aim of minimizing energy consumption. The problem faced by train dispatchers is well-known as the real-time traffic management problem, and the problem encountered by train drivers is the so-called train control problem. In fact, significant interconnections exist between these two problems, as the traffic-related properties have impact on the train-related properties, and vice versa. Solving the two problems in a sequential way hides the potential improvements in performance of train operations. Better train operations can be potentially achieved by jointly considering the two problems, i.e., (re-)constructing a train timetable in a way that applies different diving actions. However, such a joint consideration leads to a very complex and difficult optimization problem, because not only the timetable should be well-defined for synchronizing the accelerating and braking actions of trains in the same block section, but also the driving actions should be controlled under the speed limits, travel time, and distance constraints (Tuyttens et al. 2013). This is even more critical and difficult for real-time operations. Moreover, the safety headway between two consecutive trains dynamically depends on their real speed and acceleration/deceleration rate. As a result, a prompt and reliable decision-making support tool for both dispatchers and drivers is desired, which requires the integration of a rescheduling optimization with microscopic details and highly accurate real-time train speed trajectory optimization at once.

A growing body of scientific literature is available for real-time traffic management (e.g., the recent survey by Fang et al. 2015) and train control (e.g., the recent review by Yang et al. 2016). These two problems are well-studied separately, but a gap still exists with regards to their integration. Most approaches focus only on one side of the problem and include parts of the other by control loops, extra constraints, hierarchical decomposition, or additional objectives. Such focus on a single side of the problem leaves an open gap in terms of operational performance of jointly considering those two perspectives at once. The purpose of achieving better train operation and the gap in the scientific literature motivate us to address their integration.

We therefore address the integration of real-time traffic management and train control by using optimization methods, identifying both traffic-related properties (i.e., a set of times, orders, routes to be followed by trains) and train-related properties (i.e., speed trajectories) at once. To formulate the integrated problem, a mixed-integer non-linear programming (MINLP) problem ($\mathrm{P_{NLP}}$) is first proposed and solved by a two-level approach. An approximation based on piecewise affine functions, is applied to the nonlinear terms in the $\mathrm{P_{NLP}}$ problem, which results in a mixed-integer linear programming (MILP) problem ($\mathrm{P_{PWA}}$). Furthermore, a preprocessing method for generating the possible train speed profile options (TSPOs) for each train on each block section is considered to reduce the complexity of the problem and to restrict the search only to a subset that allows better energy performance. An MILP problem ($\mathrm{P_{TSPO}}$) is developed to determine the optimal option with minimum train delays.The two MILP problems are both solved by using an MILP solver, but a custom-designed two-step method is particularly used for the $\mathrm{P_{TSPO}}$ problem to speed up the solving procedure. In our optimization problems, the blocking time of a train on a block section dynamically depends on its real speed. We consider the minimization of the total train delay times as the objective. According to the experimental results, the proposed approach can obtain feasible solutions (with good quality) of the integrated traffic management and train control problem for a single direction along a 50 km corridor with 9 stations and 15 trains each hour within 3-minute computation time, meanwhile the goal of reducing train delays by managing train speed can be achieved. In Part 2 of this paper, we further discuss energy-related extensions based on the proposed optimization approaches, i.e., evaluating energy consumption and computing regenerative energy utilization. With the inclusion of the energy-related aspects, we aim at both delay recovery and energy efficiency, in order to achieve energy-efficient train operation.

The remainder of this paper is organized as follows. Section 2 provides a detailed literature review on the studies addressing the real-time railway traffic management problem without considering train dynamics, and

the studies dealing with the interaction of traffic management and train control for better train operations. In Section 3, a problem statement and assumptions are given first. Then, three optimization problems formulating the integration of traffic management and train control are presented. Section 4 introduces the solution approaches for the three proposed problems, i.e., a two-level approach for solving the MINLP problem (P_{NLP}), and a custom-designed two-step method for improving the computational efficiency of the MILP problem (P_{TSPO}). Experimental results based on a real-world railway network are given in Section 5 for evaluating the performance of the proposed approaches and investigating the benefits of the integration. Finally, Section 6 ends the paper with conclusions and topics for further research.

2. Literature review

An extensive study of literature is available for real-time railway traffic management and train control. In this section, we review the state of the art for two directions: 1) real-time traffic management, where the train speed is commonly considered fixed, i.e., a constant minimum running time is given; 2) better train operations, where traffic and train control are interacting or integrated in some way.

2.1. Real-time traffic management: better train rescheduling

The real-time railway traffic management problem has been attracting much attention in the last years. Advances in scheduling theory make it possible to solve real-life train scheduling instances, in which not only departure/arrival times (Ginkel and Schöbel 2007, D'Ariano et al. 2007a), but also train orders, routes, and further operational freedom are considered as variables (e.g., Törnquist and Persson 2007, Corman et al. 2010, 2012, Meng and Zhou 2014). For more information, we direct to the review papers by Narayanaswami and Rangaraj (2011), Corman and Meng (2015), Cacchiani et al. (2014), Fang et al. (2015), and the recent book by Hansen and Pachl (2014).

To formulate the railway network topology (infrastructure), traffic situation, and traffic constraints, several approaches based on operations research techniques are available in the scientific literature. A particularly popular stream of studies considers the alternative graph model, which uses a combination of job shop and alternative graph techniques (D'Ariano et al. 2007a). In the alternative graph model, each block section is formulated as a single capacity server with further no-store constraints and blocking constraints relating to the processing over multiple adjacent block sections (D'Ariano et al. 2007a). Some studies employ the alternative graph based formulation to deal with the problem of rerouting trains by developing meta-heuristics, e.g., a Tabu Search algorithm proposed by Corman et al. (2010); considering multiple classes of running traffic (Corman et al. 2011a); determining the Pareto frontier of the bi-objective problem of reducing delays and maintaining as many passenger connections as possible (Corman et al. 2012); investigating the impact of the levels of detail and the number of operational constraints on the applicability of models, in terms of solution quality and computational efficiency (Kecman et al. 2013); and rescheduling high-speed traffic based on a quasi-moving block system, which integrates the modeling of traffic management measures and the supervision of speed, braking, and headway (Xu et al. 2017).

Another stream of studies focuses on developing macroscopic models based on an event-activity network, which allows for faster resolution and larger geographical scope. Schöbel (2007) proposed an event-activity based integer programming model to solve the delay management problem. The model was further extended to address a discrete time/cost trade-off problem of maintaining service quality and reducing passengers' inconvenience (Ginkel and Schöbel 2007); and by including headways and capacity constraints and testing multiple pre-processing heuristics in order to fix integer variables and to speed up the computation (Schachtebeck and Schöbel 2010). In their proposed models, connections are decided to be maintained or dropped by minimizing the number of missed connections, while minimizing the sum of all delays of all events. Dollevoet et al. (2012) presented an event-activity based model to address the problem of rerouting passengers in the delay management process. Zhan et al. (2015) employed the event-activity network to reschedule the operations, when a segment of a high speed railway was totally blocked without considering rerouting, aiming to minimize the number of canceled and delayed trains.

Other approaches have also been proposed for solving the same problem. Rodriguez (2007) presented two constraint programming models for the rescheduling and rerouting of trains running through a junction,

considering a fixed speed and a variable speed respectively. The latter does not consider proper speed variation dynamics, but it constrains train running times to be coherent with train braking and acceleration in the case of conflict. Törnquist and Persson (2007) described a mathematical model for rescheduling traffic to minimize the consequences of a single disturbance, which can be an infrastructure failure, a vehicle malfunction, or a personnel availability problem. Different strategies to reschedule trains were considered, such as a change to the track used by a train or a modified train order, in order to reduce computation time depending on the size of the instance. To improve the computational efficiency, a greedy heuristic approach was further developed by Törnquist (2012), based on the same formulation of the problem. The idea was to obtain reasonably good feasible solutions in a very short time and to use the rest of the predefined computation time to improve it by backtracking and reversing decisions made in the first stage. In Mu and Dessouky (2011), a simultaneous freight train routing and scheduling problem was formulated as an MILP model with macroscopic details, which was solved via heuristic procedures based on clustering trains according to their entrance time in the network. Meng and Zhou (2014) investigated the benefits of simultaneous train rerouting and rescheduling compared to sequential approaches in general rail networks. Network-wide cumulative flow variables were used to implicitly model capacity constraints, which enabled an easy problem decomposition mechanism. The decomposed sub-problems were then solved by an adapted time-dependent least-cost algorithm. Pellegrini et al. (2014) formulated an MILP model to tackle the real-time railway traffic management problem, representing the infrastructure with fine granularity, i.e., the route-lock route-release interlocking system and the route-lock sectional-release system. They studied the problem in the case of simple junctions and more complex areas, and used CPLEX to solve the model. In Pellegrini et al. (2015), a heuristic algorithm, named RECIFE-MILP, was developed based on an extended version of the MILP formulation proposed in Pellegrini et al. (2014). Samà et al. (2016) further investigated how to select the most promising train routes among all possible alternatives, through developing an ant colony optimization meta-heuristic. The most promising subset of train routes was included in the large and complex MILP determined by Pellegrini et al. (2014) and solved with the exact and heuristic approaches presented in Pellegrini et al. (2015).

Table 1 summarizes some relevant studies on the real-time traffic management problem, in terms of problem description (i.e., the level of detail, rescheduling measure), mathematical formulation (including model structure, objective, constraints, etc.) and solution algorithm, particularly focusing on the way of handling speed dynamics. From the discussion, studies tend to consider microscopic details (including signals and switches) and precise headway between trains. Moreover, these studies mostly have a common assumption that a fixed speed profile is used for each train, given a minimum running time and neglecting the dynamic change in speed profile as a consequence of the dispatching actions. Thus, any dynamics-related objectives, such as energy consumption, cannot be considered.

2.2. Interaction of traffic management and train control: better train operations

Many studies deal with controlling the train speed, with the aim of minimizing energy consumption. In the literature, the approaches mostly identify train speed profiles using very rough approximation, at least when optimizing. A general overview of the studies can be found in the review papers by Albrecht et al. (2011), Wang et al. (2011), and Yang et al. (2016).

For operations according to the schedule, there is a large corpus of research available by now that is able to compute the regimes to be used, and to optimally follow the path of minimal energy consumption, given a running time (see e.g., Howlett and Pudney 2012, Chevrier et al. 2013, Wang et al. 2013). Some studies focused on maximizing the regenerative energy utilization, (e.g., Rodrigo et al. 2013, Yang et al. 2014). Since little interaction with traffic management is considered in these studies, we do not elaborate on them in this paper. We next focus on the studies that address the interaction/integration with traffic management in some way, e.g., in a decomposed, iterative, or non-optimized manner.

A lot of inspiration comes from metro operations, which have a particular structure of very high homogeneity (see e.g., Li and Lo 2014a,b), basic autonomy from other systems, and limited, predicted interaction along a line. The usage of Automatic Train Operations and Communication-Based Train Control is the most common paradigm to achieve precise control of running traffic (Albrecht et al. 2011). The approach implemented in the Lötschberg tunnel system was described in Montigel (2009), which simulated only very

Table 1. Summary of the relevant studies on the real-time traffic management problem

Publications	Level of detail	Rescheduling measure	Model structure	Objective(s)	Solution algorithm	Consider speed management
D'Ariano et al. (2007a)	micro	rT, rO	AG-based MILP	minimize the maximum secondary delay for all trains at all visited stations	B&B, H (FCFS, FLFS)	No
Ginkel and Schöbel (2007)	macro	rT, rO	EA-based IP	minimize the sum of train delays and the weighted sum of all missed connections	H (FSFS, FRFS, FRFS-fix, FSFS-fix)	No
Rodriguez (2007)	micro	rT, rO, rR	CPM	minimize the total delays of all trains	B&B	No
Törnquist and Persson (2007)	macro	rT	MILP	minimize the total final delays of all trains; minimize the total cost associated with delays	CS based on four different dispatching strategies	No
Corman et al. (2010)	micro	rT, rO, rR	AG-based MILP	minimize the maximum consecutive delays in lexicographic order	B&B, H (tabu search)	No
Schachtebeck and Schöbel (2010)	macro	rT, rO	EA-based IP	minimize the delays and the number of missed connections	H (FSFS, FRFS, FRFS-fix, FSFS-fix)	No
Corman et al. (2011a)	micro	rT, rO	AG-based MILP	minimize the total delays of all trains along other multiple objectives	B&B, H (priority rule based, FCFS)	No
Mu and Dessouky (2011)	macro	rT, rO, rR	MILP	minimize the total delays of all trains	GHA, NSA	No
Corman et al. (2012)	micro	rT, rO	AG-based MILP	minimize the train delays and the number of missed connections	B&B, H (pareto front based)	No
Dollevoet et al. (2012)	macro	rT, rO	EA-based IP	minimize the average delay of all passengers	CS, a modified Dijkstra's algorithm	No
Törnquist (2012)	macro	rT, rO	MILP	minimize the total final delays all trains	GHA	No
Kecman et al. (2013)	macro	rT, rO	AG-based MILP	minimize the maximum consecutive delay	B&B, H (FIFO)	No
Meng and Zhou (2014)	micro	rT, rO, rR	CF-based IP	minimize the total completion time of all trains	CS, LR, H (priority rule based)	No
Pellegrini et al. (2014, 2015)	micro	rT, rO, rR	MILP	minimize the maximum or total consecutive delays	CS, H (RECIFE-MILP)	No
Zhan et al. (2015)	macro	rT, rO	EA-based MILP	minimize the number of canceled and delayed trains	CS	No
Samà et al. (2016)	micro	rT, rO, rR	MILP	minimize the total consecutive delays	CS, ACO meta-H	No
Xu et al. (2017)	micro	rT, rO	AG-based MILP	minimize the total consecutive delays; minimize the sum of the positive consecutive delays	CS	Yes optimized speed level
Luan et al. (this work)	micro	rT, rO	MILP	Part 1: minimize the sum over all trains of the mean absolute delay time at all visited stations Part 2: minimize the sum over all trains of the mean absolute delay time at all visited stations and the energy consumption for accelerating trains and overcoming resistance	CS	Yes optimized

* Symbol descriptions for Table 1: re-time (rT); re-order (rO); re-route (rR); Alternative graph (AG); Cumulative flow (CF); Event-activity network (EA); Constraint programming model (CPM); Discrete event model (DEM); Commercial solver (CS); Heuristics (H); Branch-and-bound (B&B); Greedy heuristic algorithm (GHA); Neighborhood search algorithm (NSA); First-Leave-First-Served (FLFS); First-Come-First-Served (FCFS); First-Scheduled-First-Served (FSFS); First-Rescheduled-First-Served (FRFS); FSFS with early connection fixing (FSFS-fix); FRFS with early connection fixing (FRFS-fix); Ant colony optimization (ACO); REcherche sur la Capacité des Infrastructures FErroviaires (RECIFE, in French).

few trains at a time, in terms of train and traffic characteristics. The approach has a very good performance, but it is limited to a well-defined small test case with a limited traffic volume. The optimal solution can be found by exhaustive search; however, the scalability and applicability of the approach to different situations (e.g., larger networks and heterogeneous traffic) still need to be assessed. The approach proposed by Rao et al. (2013) aimed at pushing this concept further. Some heuristic extensions of the previous work were involved to address the open issues on the scalability and applicability to general networks and heterogeneous traffic.

In the general case of delayed and rescheduled traffic, the most common approach for integrating these two problems is the sequential adjustment of the speed profile, based on a scheduling solution that approximates or neglects the train control problem, see e.g. D'Ariano et al. (2007b, 2008). In this stream, Albrecht (2009), and D'Ariano and Albrecht (2010) focused on the energy minimization problem to deliver a continuous speed profile, given a schedule. In Albrecht et al. (2013), the time windows at stations and relevant points were used to give enough room for the rescheduling problem to calculate energy-efficient speed profiles of trains. The result is optimal for energy efficiency, given the solution to the scheduling part, i.e., the passing times of trains at stations and relevant points.

Another stream of approaches includes iterative approaches that feed an optimized speed trajectory back to the scheduling model to improve traffic performance. In general, those approaches offer no guarantee of optimality in either traffic management or train control. Such approaches include the method of Mazzarello and Ottaviani (2007) for the EU project Combine, which proposed a double feedback loop architecture to determine both traffic-related and train-related properties by heuristics. A similar approach was later proposed by Lüthi (2009), which allowed the rescheduling of trains in real time and provided dynamic schedule information to drivers, so that they can adjust their speed in order to meet the required schedule. The positive feature of such approaches is that the feedback loops keep the deviations (i.e., train delays from the planned timetable) small. However, having the two models separated means a match between the objectives of the two models has to be found; typically, this may lead to extra delay introduced by speed management. Furthermore, stability, convergence, quality of the system under a closed-loop feedback control are even more difficult to quantify than a corresponding sequential one. Quaglietta et al. (2013), Corman and Quaglietta (2015) investigated and analyzed the outcome for what concerns stability and performance inherently introduced by closing control loops.

In a different research stream, Wang and Goverde (2016) presented a multiple-phase train trajectory optimization method under real-time traffic management. The train trajectory is re-calculated to track the possibly adjusted timetable, i.e., the train schedule is updated by adjusting train speed profiles. This proposed method was only applied in a case of two successive trains running on a corridor with various delays. In such cases, train control interacts with traffic management by identifying train speed profiles that match the schedule of minimal delays. The updated trajectory solutions are fed back to re-compute an improved scheduling solution by iterative approaches, without any guarantee of optimality in either traffic management or train control.

A radically different approach is to invert the hierarchy of the problems, i.e., first solving the problem of generating efficient speed profiles and then using only these in the traffic management part. This has been operationally translated into a choice of speed profiles from a finite set: a single speed profile in the case of Corman et al. (2009), apart from retiming actions; multiple speed profiles in the case of Caimi et al. (2012), including retiming. Then those profiles were included in the optimization problem. Two conflicting objectives of energy efficiency and delay minimization were considered in Corman et al. (2009), in which the first objective was used as a hard constraint. Two policies were analyzed: 1) waiting in corridors, i.e., trains are allowed to wait in stations and along the line; and 2) green-wave, where trains can wait only at stations. In Caimi et al. (2012), the retiming and rerouting decisions were combined through the definition of blocking-stairways, each one combining a routing and a speed profile, selecting then few among a finite number of alternatives for each trains.

In Zhou et al. (2017), a unified model was developed based on a space-time-speed grid network to integrate the two problems of macroscopic train timetabling and microscopic train trajectory calculations for high-speed rail lines. Most information regarding traffic properties and train properties was pre-described in the space-time-speed grid network, and the integrated problem was then simplified as a path finding problem.

A dynamic programming solution algorithm was proposed to find the train speed profile solutions with dualized train headway and power supply constraints.

2.3. Paper contribution

The vast majority of the optimization-based train rescheduling approaches has a common assumption that a fixed speed profile is used for each train, i.e., a pre-determined (constant) minimum running time for each train is considered and train dynamics are neglected, as reviewed in Section 2.1. As a result, any dynamics-related objectives, such as energy consumption, cannot be directly considered in the optimization. The studies on train control mostly focus on trajectory optimization with a given running time, i.e., determining the driving regimes and the switching points, with the aim of minimizing energy consumption (see the review paper by Yang et al. 2016). As significant correlations exist between these two problems, some studies try to address their interaction/integration in a decomposed, iterative, or non-optimized manner, refer to Section 2.2. However, few authors deal with the integrated problem by employing mathematical optimization methods. When they do so, they typically either address the energy-efficient management problem for the urban transit systems (e.g., Li and Lo 2014a,b) or the high-speed railway lines (e.g., Zhou et al. 2017) with high homogeneity, classify speed into several levels and managing speed by indicating additional travel time (e.g., Xu et al. 2017), or focus on one of these two problems with some simplification of the other (e.g., Caimi et al. 2012).

Moreover, train operations require safety separation over block sections, in terms of time headway or space headway. The safety headway, either time headway or space headway, between two consecutive trains dynamically depends on their real speed and acceleration/deceleration rate. In real operations, we cannot assume that all traffic runs in free-flow conditions. To deal with this issue, an integrated model with microscopic details is needed that is able to consider variable running times and safety headways, according to the train speed, accelerating or deceleration features.

Based on the achievements and gaps in the literature, the main contributions of this paper are summarized as follows:

- This study integrates two single optimization problem decisions on real-time traffic management and train control, which are typically addressed in a separated, decomposed, iterative, or non-optimized manner in previous studies. The integrated modeling approach is innovative, and it incorporates the representations of microscopic traffic regulations and train speed trajectories into a single optimization model.
- An MINLP model and two MILP models are proposed to construct the real-time train timetable in a way of optimizing the train accelerating and braking actions. A train dispatching solution and a train control solution are delivered at once by each proposed model.
- In our models, the train speed is considered to be variable, whereas it is commonly assumed to be constant and regarded as a minimum running time in previous traffic management research. The blocking time dynamically depends on the real operating train speed, and we do not assume a fixed minimum safety headway anymore.
- Comprehensive experiments are conducted based on a real-world railway network with heterogeneous traffic[1], which is relatively complex in comparison with most existing studies of the integrated problem in the literature. An analysis of the experimental results identifies the good/satisfactory performance of the P_{TSPO} model and the potential benefits of the integration. The performance of the proposed approaches on larger-scale networks is also examined on a more sophisticated railway network from INFORMS RAS (2012).
- The proposed methods enable us to identify and evaluate the performance-related indicators. Compared with the solutions neglecting train dynamics, the solutions obtained by using the proposed methods achieve up to 8% reduction of train delay.

[1] In a heterogeneous situation, trains with different speeds or different stop patterns may interfere with each other.

3. Mathematical formulations

In this section, after a problem statement and formulation assumptions, three optimization approaches are proposed to address the integration of traffic management and train control, i.e., an MINLP approach (P_{NLP}) presented in Section 3.2.1, an MILP approach (P_{PWA}) obtained by approximating the nonlinear terms with PWA functions in Section 3.2.2, and another MILP approach (P_{TSPO}) considering multiple TSPOs generated in a preprocessing step (Section 3.2.3).

3.1. Problem statement and formulation assumptions

The safety headway time is the time interval between two following trains and the minimum headway depends on the so-called "blocking time" (Pachl 2009). The blocking time is the time interval in which a section of track (usually a block section) is exclusively allocated to a train and therefore blocked for other trains. Thus, the blocking time lasts from the moment of issuing a train movement authorization (e.g., by clearing a signal) to the moment that it becomes possible to issue a movement authorization to another train to enter that same section. The blocking time of a block section is usually much longer than the time that the train occupies the block section. Fig. 1(a) and Fig. 1(b) illustrate the blocking time of a block section for a train without and with a scheduled stop respectively.

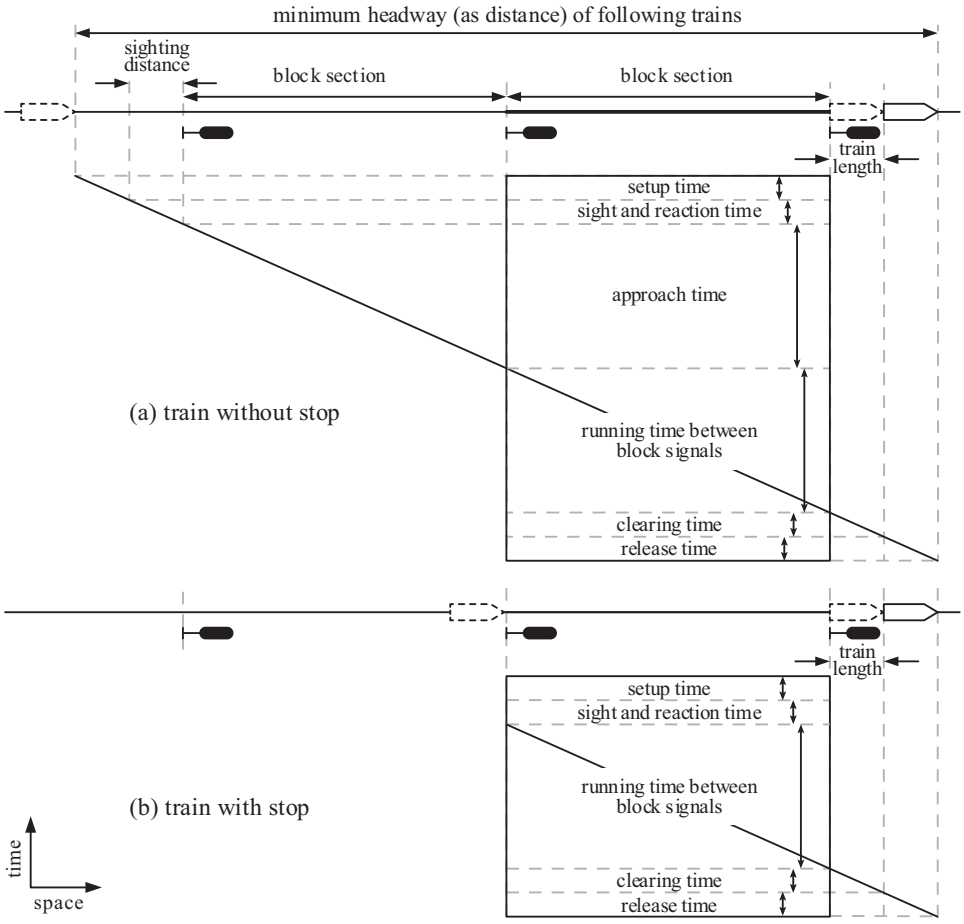


Fig. 1. The blocking time of a block section for a train without/with a scheduled stop

Pachl (2009) defined the components of the blocking time illustrated in Fig. 1 as follows: 1) the *setup time* is the time duration for clearing the signal before the arrival of a train; 2) the *sight and reaction time*

is a certain time duration for the driver to view the signal; 3) the *approach time* is the time duration for train running over the preceding block section (from the approach signal to the block signal); 4) the *running time* is the time duration for a train to run on the block section; 5) the *clearing time* is the time duration to clear the block section and the overlap with the full length of the train (if required) after the departure of a train; 6) the *release time* is used to unlock the safety block system. Note that the five components of the blocking time are all time durations, the former three terms are used for pre-blocking a block section, and the latter two terms are for post-releasing a block section. Based on the explanations, the approach time and the clearing time strongly depend on the train characteristics (e.g., train speed and train length) and the rail network conditions (e.g., length of block section); therefore, they are considered as decision variables and the others (e.g., the setup time) are regarded as constant in this paper.

Given a railway network with the technical and operational requirements of stations and segments (e.g., lengths of block section, speed limitations, and allowed/un-allowed dwelling events), a set of trains from pre-specified origins to pre-specified destinations and with pre-specified train characteristics (e.g., length, speed limitation, acceleration, and deceleration), the statement of the integrated traffic management and train control problem is to determine the routes, orders, arrival times, and departure times of the trains at passing stations by finding the optimal train speed profiles, in order to reduce the train delay, and at the same time save the energy for accelerating and re-accelerating caused by unnecessary braking.

We focus on the investigation of the traffic operations. Thus, when constructing the formulations, we emphasize in detail the operational aspect of the traffic and consider the train control aspect with relatively less accuracy in computing the energy consumption (at least, compared with the studies only focusing on train trajectory optimization). In fact, what we target is not to take decisions to change the cruising speed of trains (as it may result in lots of delays due to the high dependence among trains), or to exploit running time buffers to save energy (which can be done focusing on a single train at a time only, running ahead of time), but mostly by avoiding unnecessary acceleration and deceleration due to interaction of traffic. We construct and reschedule the train timetable by optimizing the train accelerating and braking actions. Therefore, in our optimization problems, we make the following assumptions: (1) train acceleration is considered as a piecewise constant function by giving a fixed switching point (breakpoint) of speed (e.g., 60 km/h) for each train category; (2) train deceleration is constant for a certain train category and differs among train categories; (3) the speed limit is considered as constant for a certain train category on a certain block section, i.e., the minimum value of the designed train speed and the designed block section (track) speed, but differs among train categories and block sections; (4) the beginning/ending point of a block section or of a main/siding track in a station, or a point of merging/diverging of tracks on a segment, is represented by a node; (5) a block section is described as a cell, which connects two nodes in a pair; (6) a station is simplified to a number of main/siding track(s), which can be further modeled as a single cell or a set of cells; (7) for a double-track railway segment between two stations, each track is modeled as a sequence of directional cells (i.e., directional block sections), and for a single-track railway segment, the only track between two stations is modeled as bi-directional cells (i.e., bi-directional block section); (8) the speed of a train on a cell is divided into three phases, i.e., incoming, cruising, and outgoing phases, and train coasting is neglected (however, a coasting phase can be introduced by assuming a piecewise constant deceleration function of the cruising speed, as discussed in Part 2 of this paper); (9) the resistances caused by air, roll, track grade, curves, and tunnels are not considered in this part, but they are included in Part 2 while evaluating energy consumption, i.e., the energy consumed for overcoming resistance in accelerating, cruising, and decelerating is computed in Part 2; (10) only one train is allowed to access a cell at any time; (11) the time step (granularity of time) is one second. Note that the maximum acceleration and deceleration depend on the traction and braking force. In the literature, the researchers either consider tractive force as a precise function of speed and control (Howlett 2000), or assume constant power (then tractive force is a function of speed, e.g., Howlett 2000), or assume to have constant acceleration (Wang et al. 2016).

3.2. Three mathematical formulations for integrating the traffic management and train control

3.2.1. Formulation of the $\mathrm{P_{NLP}}$ *problem*

Table 2 lists the sets, subscripts, input parameters, and decision variables used by the $\mathrm{P_{NLP}}$ problem.

Table 2. Sets, subscripts, input parameters, and decision variables

Symbol	Description
	Sets and subscripts
F	set of trains, $\|F\|$ is the number of trains
V	set of nodes, $\|V\|$ is the number of nodes
E	set of cells, i.e., block sections, $E \subseteq V \times V$, $\|E\|$ is the number of cells
f	train index, $f \in F$
p, i, j, k	node index, $p, i, j, k \in V$
e	cell index, denoted by (i, j), $e \in E$
E_f	set of cells (or sections) that train f may use, $E_f \subseteq E$
E_f^{stop}	set of cells in which train f should stop, $E_f^{\text{stop}} \subseteq E_f$, $\|E_f^{\text{stop}}\|$ is the number of stops of train f
	Input parameters
o_f/s_f	origin/destination node of train f
L_f^{train}	length of train f
c_f^{pri}	primary delay time of train f at its origin node
c_f	planned departure time of train f at its origin node
ρ_f	direction of train f
v_f^{turn}	the train speed at the switching point of acceleration for train f
v^{mincru}	the minimum cruising speed for each train on each cell
v_i^{nlim}	train speed limitation at node i
$v_{i,j}^{\text{clim}}$	train speed limitation on cell (i, j)
$L_{i,j}^{\text{cell}}$	length of cell (i, j)
$D_{f,i,j}$	planned arrival time of train f on cell (i, j), $(i, j) \in E_f^{\text{stop}}$
$w_{f,i,j}^{\min}/w_{f,i,j}^{\max}$	the minimum/maximum dwell time of train f on cell (i, j)
$\alpha_{1,f,i,j}$	the maximum acceleration of train f on cell (i, j), when train speed is not larger than v_f^{turn}
$\alpha_{2,f,i,j}$	the maximum acceleration of train f on cell (i, j), when train speed is larger than v_f^{turn}
$\beta_{f,i,j}$	the maximum deceleration of train f on cell (i, j)
$\tau_{f,i,j}^{\text{setup}}$	setup time for setting cell (i, j) when train$f$ is approaching
$\tau_{f,i,j}^{\text{sight}}$	sight time, i.e., running time over a sight distance when train f is approaching cell (p, i). Note that cell (p, i) is the preceding cell of cell (i, j)
$\tau_{f,i,j}^{\text{reaction}}$	reaction time of train f's driver while approaching cell (i, j)
$\tau_{f,i,j}^{\text{release}}$	release time for releasing cell (i, j) after the clearance of train f
M/ϵ	a sufficiently large/small positive number
	Decision variables
$a_{f,i,j}/d_{f,i,j}$	arrival/departure time of train f at cell (i, j)
$a_{f,i,j}^{\text{turn}}/d_{f,i,j}^{\text{turn}}$	time point that train f reaches the switching speed v_f^{turn} in the incoming/outgoing phase on cell (i, j)
$a_{f,i,j}^{\text{cru}}/d_{f,i,j}^{\text{cru}}$	time point that train f starts/ends cruising, i.e., the starting/ending time of cruising phase on cell (i, j)
$v_{f,i,j}^{\text{in}}/v_{f,i,j}^{\text{cru}}/$ $v_{f,i,j}^{\text{out}}$	incoming speed, cruising speed, and outgoing speed of train f on cell (i, j)
$\theta_{f,f',i,j}$	binary train ordering variables, $\theta_{f,f',i,j} = 1$ if train f' arrives at cell (i, j) after train f, and otherwise $\theta_{f,f',i,j} = 0$
$w_{f,i,j}$	dwell time of train f on cell (i, j)
$\tau_{f,i,j}^{\text{approach}}$	approach time of train f on cell (i, j), i.e., running time of train f on the preceding cell (p, i)
$\tau_{f,i,j}^{\text{clear}}$	clearing time for clearing cell (i, j) with the length of train f
$g_{f,i,j}$	safety time interval between occupancy of cell (i, j) and arrival of train f

Symbol	Description
$h_{f,i,j}$	safety time interval between departure of train f and release of cell (i,j)
$\sigma_{f,i,j}/\delta_{f,i,j}$	occupancy/release time of cell (i,j) for train f
$\Theta^{\text{in}}_{f,i,j}/\Theta^{\text{out}}_{f,i,j}$	energy consumption of train f caused by traction force, represented by the difference of the squared speeds in the incoming/outgoing phase on cell (i,j)
$\zeta_{1,f,i,j},\ldots,$ $\zeta_{6,f,i,j}$	logical variables to indicate the relation of the incoming, cruising, outgoing speed, and switching speed v^{turn}_f, for train f on cell (i,j), as explained in Table 3

Three types of variables are used to formalize the traffic and train related decisions: time variables a and d, speed variables v, and train order variables θ. The other variables are a consequence of the interactions among these variables for all trains in the network, with respect to the formulas of the uniformly accelerating and decelerating motions, definition of the blocking time, and safety requirements.

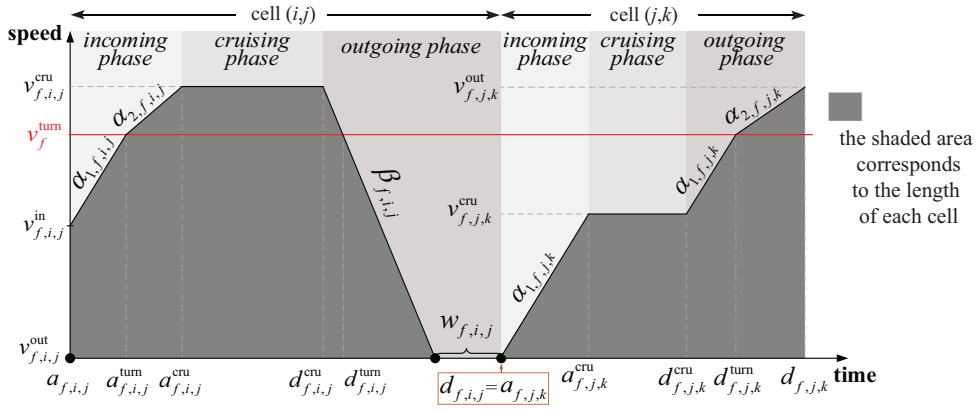


Fig. 2. Speed-time graph of train f on cell (i,j) and cell (j,k) to illustrate the relevant decision variables

Fig. 2 illustrates the relevant variables of train f on two adjacent cells, namely cell (i,j) and cell (j,k). The trajectory of train f on each cell is divided into three phases: incoming, cruising, and outgoing phases. As illustrated in Fig. 2, train f enters cell (i,j) at time $a_{f,i,j}$ with a speed $v^{\text{in}}_{f,i,j}$, and then a sequence of the following actions is taken on cell (i,j):

1) in the time interval $[a_{f,i,j}, a^{\text{turn}}_{f,i,j}]$, the train accelerates from speed $v^{\text{in}}_{f,i,j}$ to speed v^{turn}_f at a steady acceleration $\alpha_{1,f,i,j}$;
2) in the time interval $[a^{\text{turn}}_{f,i,j}, a^{\text{cru}}_{f,i,j}]$, the train accelerates from speed v^{turn}_f to speed $v^{\text{cru}}_{f,i,j}$ at a steady acceleration $\alpha_{2,f,i,j}$;
3) in the time interval $[a^{\text{cru}}_{f,i,j}, d^{\text{cru}}_{f,i,j}]$, the train keeps a constant speed $v^{\text{cru}}_{f,i,j}$;
4) in the time interval $[d^{\text{cru}}_{f,i,j}, d_{f,i,j} - w_{f,i,j}]$, the train decelerates from speed $v^{\text{cru}}_{f,i,j}$ to speed $v^{\text{out}}_{f,i,j}$ (i.e., 0 km/h in this case) at a steady deceleration $-\beta_{f,i,j}$;
5) in the time interval $[d_{f,i,j} - w_{f,i,j}, d_{f,i,j}]$, the train dwells in cell (i,j).

Then, train f departs from cell (i,j) at time $d_{f,i,j}$. Meanwhile, train f arrives at cell (j,k) at time $a_{f,j,k}$, and starts accelerating. As train f does not reach the switching speed v^{turn}_f in the incoming phase of cell (j,k), only one acceleration $\alpha_{1,f,i,j}$ is used. Note that the sequence of the action(s) taken by a train on a cell do not follow a pre-specified frame (like the one described above); in fact, it is determined by optimizing the time variables (a/d) and speed variables (v). For instance, a train may take a sequence of actions to first accelerate and then decelerate on a cell (i.e., $v^{\text{in}}_{f,i,j} < v^{\text{cru}}_{f,i,j}$ and $v^{\text{out}}_{f,i,j} < v^{\text{cru}}_{f,i,j}$), and it may also take only one action to keep a constant speed traversing the cell (i.e., $v^{\text{in}}_{f,i,j} = v^{\text{cru}}_{f,i,j} = v^{\text{out}}_{f,i,j}$). All possible train trajectories in the incoming and outgoing phases are intuitively provided and explained in Table 6 of Appendix A.1.

We next formulate the integrated traffic management and train control problem. As commonly used in train dispatching optimization problems, each train is assigned a planned arrival time at each planned stop. In the objective function, we minimize the sum over all trains of the mean absolute delay time at all visited stations, i.e., minimizing the deviation from the planned train timetable:

$$\min Z = \sum_{f \in F} \sum_{(i,j) \in E_f^{\text{stop}}} \frac{|d_{f,i,j} - w_{f,i,j} - D_{f,i,j}|}{\left|E_f^{\text{stop}}\right|}, \tag{1}$$

The train speed consistency constraint

$$v_{f,i,j}^{\text{out}} = v_{f,j,k}^{\text{in}}, \quad \forall f \in F, j \neq o_f, (i,j) \in E_f, (j,k) \in E_f \tag{2}$$

ensures the consistency of the train speed between two adjacent cells, i.e., the incoming speed of train f on cell (j,k) equals to its outgoing speed on the preceding cell (i,j).

A set of train speed limitation constraints is presented, in which

$$v_{f,o_f,j}^{\text{in}} = 0, \quad \forall f \in F, (o_f, j) \in E_f, \tag{3}$$

$$v_{f,j,s_f}^{\text{out}} = 0, \quad \forall f \in F, (j, s_f) \in E_f \tag{4}$$

guarantee that trains stop at their origins and destinations respectively, i.e., the incoming speed of the origin cell (o_f, j) and the outgoing speed of the destination cell (j, s_f) is zero, and

$$0 \leq v_{f,i,j}^{\text{in}} \leq v_i^{\text{nlim}}, \quad \forall f \in F, (i,j) \in E_f, \tag{5}$$

$$0 \leq v_{f,i,j}^{\text{out}} \leq v_j^{\text{nlim}}, \quad \forall f \in F, (i,j) \in E_f, \tag{6}$$

$$v^{\text{mincru}} \leq v_{f,i,j}^{\text{cru}} \leq v_{i,j}^{\text{clim}}, \quad \forall f \in F, (i,j) \in E_f \tag{7}$$

ensure that train speed cannot exceed the given speed limitation at each node and on each cell.

The following constraint

$$a_{f,i,j} \leq a_{f,i,j}^{\text{turn}} \leq a_{f,i,j}^{\text{cru}} \leq d_{f,i,j}^{\text{cru}} \leq d_{f,i,j}^{\text{turn}} \leq d_{f,i,j} - w_{f,i,j}, \quad \forall f \in F, (i,j) \in E_f \tag{8}$$

ensures a proper sequence of the multiple events of train f on cell (i,j), e.g., train arrival, cruising, and departure occur in sequence.

The cell-to-cell transition constraint

$$d_{f,i,j} = a_{f,j,k}, \quad \forall f \in F, (i,j) \in E_f, (j,k) \in E_f \tag{9}$$

enforces the transition time between two adjacent cells, i.e., the departure time of train f on the preceding cell (i,j) equals the arrival time of train f on the successive cell (j,k), if two adjacent cells (i,j) and (j,k) are used consecutively by train f.

The earliest departure time constraint

$$a_{f,o_f,j} \geq c_f + c_f^{\text{pri}}, \quad \forall f \in F, (o_f, j) \in E_f \tag{10}$$

ensures that trains do not leave their origins before the earliest departure time, i.e., the sum of the planned departure time and the primary delay time.

A set of train dwell time constraints is considered, in which

$$w_{f,i,j}^{\min} \leq w_{f,i,j} \leq w_{f,i,j}^{\max}, \quad \forall f \in F, (i,j) \in E_f \tag{11}$$

guarantees the required minimum and maximum dwell times at stations, and

$$\begin{cases} v_{f,i,j}^{\text{out}} = 0, & \text{if } w_{f,i,j} > 0 \\ v_{f,i,j}^{\text{out}} > 0, & \text{if } w_{f,i,j} = 0 \end{cases}, \quad \forall f \in F, (i,j) \in E_f \tag{12}$$

links the outgoing speed variables $v_{f,i,j}^{\text{out}}$ and the dwell time variables $w_{f,i,j}$. The maximum dwell time is used to avoid un-allowed dwell events of trains. If a train is allowed to stop on a block section (in a general case), then the corresponding maximum dwell time is set to be sufficiently large; if a train is required to not stop on some particular block sections, then the maximum dwell times on these particular block sections are set to be zero. In (12), if train f stops on cell (i,j), i.e., the dwell time $w_{f,i,j}$ is larger than zero, then the corresponding outgoing speed $v_{f,i,j}^{\text{out}}$ equals zero; otherwise, $v_{f,i,j}^{\text{out}}$ should be larger than zero. Note that

constraint (12) is an "if-then" constraint, which can be rewritten as mixed-integer linear constraints by applying the transformation properties in Williams (2013), which will be introduced in Section 3.2.2.

The cell length constraints can be written as

$$L_{i,j}^{\text{cell}} = L_{f,i,j}^{\text{in}} + L_{f,i,j}^{\text{cru}} + L_{f,i,j}^{\text{out}}, \quad \forall f \in F, (i,j) \in E_f, \tag{13}$$

where $L_{f,i,j}^{\text{in}}$, $L_{f,i,j}^{\text{cru}}$, and $L_{f,i,j}^{\text{out}}$ indicate the distance that train f runs through on cell (i,j) in the incoming, cruising, and outgoing phases respectively; these distances are given by the following equations:

$$L_{f,i,j}^{\text{in}} = \begin{cases} \frac{1}{2}\left(v_{f,i,j}^{\text{in}} + v_f^{\text{turn}}\right)\left(a_{f,i,j}^{\text{turn}} - a_{f,i,j}\right) + \frac{1}{2}\left(v_f^{\text{turn}} + v_{f,i,j}^{\text{cru}}\right)\left(a_{f,i,j}^{\text{cru}} - a_{f,i,j}^{\text{turn}}\right), \text{ if } v_{f,i,j}^{\text{in}} \le v_f^{\text{turn}} \le v_{f,i,j}^{\text{cru}} \\ \frac{1}{2}\left(v_{f,i,j}^{\text{in}} + v_{f,i,j}^{\text{cru}}\right)\left(a_{f,i,j}^{\text{cru}} - a_{f,i,j}\right), \text{ otherwise} \end{cases} \tag{14a}$$

$$L_{f,i,j}^{\text{cru}} = v_{f,i,j}^{\text{cru}} \cdot \left(d_{f,i,j}^{\text{cru}} - a_{f,i,j}^{\text{cru}}\right), \tag{14b}$$

$$L_{f,i,j}^{\text{out}} = \begin{cases} \frac{1}{2}\left(v_{f,i,j}^{\text{cru}} + v_f^{\text{turn}}\right)\left(d_{f,i,j}^{\text{turn}} - d_{f,i,j}^{\text{cru}}\right) + \frac{1}{2}\left(v_f^{\text{turn}} + v_{f,i,j}^{\text{out}}\right)\left(d_{f,i,j} - w_{f,i,j} - d_{f,i,j}^{\text{turn}}\right), \text{ if } v_{f,i,j}^{\text{cru}} \le v_f^{\text{turn}} \le v_{f,i,j}^{\text{out}} \\ \frac{1}{2}\left(v_{f,i,j}^{\text{cru}} + v_{f,i,j}^{\text{out}}\right)\left(d_{f,i,j} - w_{f,i,j} - d_{f,i,j}^{\text{cru}}\right), \text{ otherwise} \end{cases} \tag{14c}$$

These equations derive from the basic formulas of uniformly accelerating or decelerating motions, i.e., for such a motion with an initial speed v_o, a final speed v_t and an elapsed time Δt, the distance traveled is $L = \frac{v_0+v_t}{2} \cdot \Delta t$. Note that the distance $L_{i,j}^{\text{in}}$ that train f runs over on cell (i,j) equals the length of cell (i,j) and corresponds to the shaded area in Fig. 2. Constraints (14a)-(14c) are nonlinear, due to the nonlinear dynamics of time, speed, and distance.

The approach time and clearing time constraints can be written as

$$\tau_{f,j,k}^{\text{approach}} = \begin{cases} 0, & \text{if } w_{f,i,j} > 0 \\ d_{f,i,j} - a_{f,i,j}, & \text{if } w_{f,i,j} = 0 \end{cases}, \quad \forall f \in F, (i,j) \in E_f, (j,k) \in E_f, \tag{15}$$

$$\tau_{f,p,i}^{\text{clear}} = 2 \cdot L_f^{\text{train}} / (v_{f,p,i}^{\text{out}} + v_{f,i,j}^{\text{cru}}), \quad \forall f \in F, (p,i) \in E_f, (i,j) \in E_f. \tag{16}$$

These two constraints are also nonlinear. In (15), if train f does not stop on the preceding cell (i,j), the approach time of train f on cell (j,k) equals its running time on the preceding cell (i,j); otherwise, the approach time of train f on cell (j,k) equals zero. This "if-then" constraint can be rewritten as mixed-integer linear constraints by applying the transformation properties in Williams (2013), which will be introduced in Section 3.2.2. The clearing time of train f on cell (p,i) is determined in (16) according to its incoming and cruising speed on the successive cell (i,j).

A set of equations is proposed for determining the safety time interval illustrated in Fig. 1, in which:

$$g_{f,i,j} = \tau_{f,i,j}^{\text{setup}} + \tau_{f,i,j}^{\text{sight}} + \tau_{f,i,j}^{\text{reaction}} + \tau_{f,i,j}^{\text{approach}}, \quad \forall f \in F, (i,j) \in E_f \tag{17}$$

defines the safety time interval between cell occupancy and train arrival, including the setup time $\tau_{f,i,j}^{\text{setup}}$, the sight time $\tau_{f,i,j}^{\text{sight}}$, the reaction time $\tau_{f,i,j}^{\text{reaction}}$, and the approach time $\tau_{f,i,j}^{\text{approach}}$, and

$$h_{f,i,j} = \tau_{f,i,j}^{\text{release}} + \tau_{f,i,j}^{\text{clear}}, \quad \forall f \in F, (i,j) \in E_f \tag{18}$$

calculates the safety time interval between train departure and cell release, including the release time $\tau_{f,i,j}^{\text{release}}$ and the clearing time $\tau_{f,i,j}^{\text{clearing}}$.

Then, the cell occupancy and cell release times, i.e., the blocking time for train f traversing cell (i,j), can be respectively written as

$$\sigma_{f,i,j} = a_{f,i,j} - g_{f,i,j}, \quad \forall f \in F, (i,j) \in E_f, \tag{19}$$

$$\delta_{f,i,j} = d_{f,i,j} + h_{f,i,j}, \quad \forall f \in F, (i,j) \in E_f. \tag{20}$$

The following constraint

$$\theta_{f,f',i,j} + \theta_{f',f,i,j} = 1, \quad \forall f \in F, f' \in F, (i,j) \in E_f, (i,j) \in E_{f'} \tag{21}$$

indicates that either train f' arrives at cell (i,j) after train f or train f arrives at cell (i,j) after train f'.

Recall that as cells can be bi-directional, trains can use the same cell in different directions, i.e., it is possible to use cell (i,j) and (j,i). Based on the restriction of the train orders in (21), the cell capacity

constraints can be written as

$$\sigma_{f',i,j} + (1 - \theta_{f,f',i,j}) \cdot M \geq \delta_{f,i,j}, \quad \forall f \in F, f' \in F, f \neq f', \rho_f = \rho_{f'}, (i,j) \in E_f, (i,j) \in E_{f'}, \tag{22}$$

$$\sigma_{f',j,i} + (1 - \theta_{f,f',i,j}) \cdot M \geq \delta_{f,i,j}, \quad \forall f \in F, f' \in F, f \neq f', \rho_f \neq \rho_{f'}, (i,j) \in E_f, (j,i) \in E_{f'}. \tag{23}$$

Constraints (22) and (23) ensure that any pair of trains using one cell in the same or different direction respectively are conflict-free, by avoiding the overlap between the cell release time for a preceding train and the cell occupancy time for a successive train. Specifically, for both train f and f' traversing cell (i,j) (i.e., with the same running direction $\rho_f = \rho_{f'}$), if train f' arrives at cell (i,j) after train f, i.e., $\theta_{f,f',i,j} = 1$, constraint (23) is non-active and (22) reduces to $\sigma_{f',i,j} \geq \delta_{f,i,j}$, which implies that the occupancy time of cell (i,j) for train f' should be later than the release time of cell (i,j) for train f.

Table 3. Explanation of the speed indicators $\zeta_{1,f,i,j}, ..., \zeta_{6,f,i,j}$ for train f on cell (i,j)

	Incoming phase			Outgoing phase		
Speed conditions	$v^{\text{in}}_{f,i,j} \leq v^{\text{cru}}_{f,i,j}$	$v^{\text{turn}}_{f} \leq v^{\text{in}}_{f,i,j}$	$v^{\text{cru}}_{f,i,j} \leq v^{\text{turn}}_{f}$	$v^{\text{cru}}_{f,i,j} \leq v^{\text{out}}_{f,i,j}$	$v^{\text{turn}}_{f} \leq v^{\text{cru}}_{f,i,j}$	$v^{\text{out}}_{f,i,j} \leq v^{\text{turn}}_{f}$
	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$
Speed indicators	$\zeta_{1,f,i,j} = 1$	$\zeta_{3,f,i,j} = 1$	$\zeta_{4,f,i,j} = 1$	$\zeta_{2,f,i,j} = 1$	$\zeta_{5,f,i,j} = 1$	$\zeta_{6,f,i,j} = 1$

To formulate the uniformly accelerating and decelerating motions, six logical speed indicators $\zeta_{1,f,i,j}$, ..., $\zeta_{6,f,i,j}$ are used to indicate the train speed. Table 3 gives an overview of the link between the speed conditions and the speed indicators, and Appendix A.1 provides the detailed explanation of these indicators. By adapting the transformation properties in Williams (2013) (briefly introduced in Section 3.2.2), these if-then constraints can be further represented by a set of linear inequalities. For instance, $\zeta_{1,f,i,j} = 1$, if and only if $v^{\text{in}}_{f,i,j} \leq v^{\text{cru}}_{f,i,j}$ can be represented by the following inequalities:

$$v^{\text{in}}_{f,i,j} - v^{\text{cru}}_{f,i,j} \leq v^{\text{nlim}}_{i} \cdot \left(1 - \zeta_{1,f,i,j}\right), \tag{24a}$$

$$v^{\text{in}}_{f,i,j} - v^{\text{cru}}_{f,i,j} \geq \varepsilon + \left(-v^{\text{clim}}_{i,j} - \varepsilon\right) \cdot \zeta_{1,f,i,j}, \tag{24b}$$

where v^{nlim}_{i} is the upper bound of $(v^{\text{in}}_{f,i,j} - v^{\text{cru}}_{f,i,j})$ and $-v^{\text{clim}}_{i,j}$ is the lower bound of $(v^{\text{in}}_{f,i,j} - v^{\text{cru}}_{f,i,j})$.

Thanks to the logical speed indicators $\zeta_{1,f,i,j}, ..., \zeta_{6,f,i,j}$, we can formulate the uniformly accelerating and decelerating motion in a linear manner and consider multiple scenarios (in which different values of acceleration and deceleration are required) at once. The following set of constraints is presented for the incoming phase, in which

$$-\tfrac{v^{\text{cru}}_{f,i,j} - v^{\text{in}}_{f,i,j}}{\beta_{f,i,j}} - M \cdot \zeta_{1,f,i,j} \leq a^{\text{cru}}_{f,i,j} - a_{f,i,j} \leq -\tfrac{v^{\text{cru}}_{f,i,j} - v^{\text{in}}_{f,i,j}}{\beta_{f,i,j}} + M \cdot \zeta_{1,f,i,j} \tag{25a}$$

indicates the uniformly decelerating motion at a steady deceleration $-\beta_{f,i,j}$,

$$\tfrac{v^{\text{cru}}_{f,i,j} - v^{\text{in}}_{f,i,j}}{\alpha_{2,f,i,j}} - M \cdot (2 - \zeta_{1,f,i,j} - \zeta_{3,f,i,j}) \leq a^{\text{cru}}_{f,i,j} - a_{f,i,j} \leq \tfrac{v^{\text{cru}}_{f,i,j} - v^{\text{in}}_{f,i,j}}{\alpha_{2,f,i,j}} + M \cdot (2 - \zeta_{1,f,i,j} - \zeta_{3,f,i,j}) \tag{25b}$$

indicates the uniformly accelerating motion at a steady acceleration $\alpha_{2,f,i,j}$, when the train speed is always larger than the switching speed v^{turn}_{f},

$$\tfrac{v^{\text{cru}}_{f,i,j} - v^{\text{in}}_{f,i,j}}{\alpha_{1,f,i,j}} - M \cdot (2 - \zeta_{1,f,i,j} - \zeta_{4,f,i,j}) \leq a^{\text{cru}}_{f,i,j} - a_{f,i,j} \leq \tfrac{v^{\text{cru}}_{f,i,j} - v^{\text{in}}_{f,i,j}}{\alpha_{1,f,i,j}} + M \cdot (2 - \zeta_{1,f,i,j} - \zeta_{4,f,i,j}) \tag{25c}$$

indicates the uniformly accelerating motion at a steady acceleration $\alpha_{1,f,i,j}$, when the train speed is always less than the switching speed v^{turn}_{f}, and

$$\begin{aligned} \tfrac{v^{\text{turn}}_{f} - v^{\text{in}}_{f,i,j}}{\alpha_{1,f,i,j}} - M \cdot (1 - \zeta_{1,f,i,j} + 2 \cdot \zeta_{3,f,i,j} + 2 \cdot \zeta_{4,f,i,j}) &\leq a^{\text{turn}}_{f,i,j} - a_{f,i,j} \\ &\leq \tfrac{v^{\text{turn}}_{f} - v^{\text{in}}_{f,i,j}}{\alpha_{1,f,i,j}} + M \cdot (1 - \zeta_{1,f,i,j} + 2 \cdot \zeta_{3,f,i,j} + 2 \cdot \zeta_{4,f,i,j}) \end{aligned} \tag{25d}$$

$$\begin{aligned} \tfrac{v^{\text{cru}}_{f,i,j} - v^{\text{turn}}_{f}}{\alpha_{2,f,i,j}} - M \cdot (1 - \zeta_{1,f,i,j} + 2 \cdot \zeta_{3,f,i,j} + 2 \cdot \zeta_{4,f,i,j}) &\leq a^{\text{cru}}_{f,i,j} - a^{\text{turn}}_{f,i,j} \\ &\leq \tfrac{v^{\text{cru}}_{f,i,j} - v^{\text{turn}}_{f}}{\alpha_{2,f,i,j}} + M \cdot (1 - \zeta_{1,f,i,j} + 2 \cdot \zeta_{3,f,i,j} + 2 \cdot \zeta_{4,f,i,j}) \end{aligned} \tag{25e}$$

indicate a two-stage uniformly accelerating motion, i.e., the train first accelerates at a steady acceleration $\alpha_{1,f,i,j}$ and then accelerates at a steady acceleration $\alpha_{2,f,i,j}$. The detailed explanation of (25) is provided in Appendix A.2.

To compute the time points $a^{\text{turn}}_{f,i,j}$ and $d^{\text{turn}}_{f,i,j}$ under some special scenarios, e.g., a train does not reach the switching speed v^{turn}_f on a cell, the following set of constraints is proposed for the incoming phase:

$$a^{\text{turn}}_{f,i,j} \leq a_{f,i,j} + M \cdot |\zeta_{1,f,i,j} - \zeta_{3,f,i,j}|\,, \tag{26a}$$

$$a^{\text{turn}}_{f,i,j} \geq a^{\text{cru}}_{f,i,j} - M \cdot |\zeta_{1,f,i,j} - \zeta_{4,f,i,j}|\,. \tag{26b}$$

Specifically, when $\zeta_{1,f,i,j} = \zeta_{3,f,i,j}$, i.e., $v^{\text{turn}}_f \leq v^{\text{in}}_{f,i,j} \leq v^{\text{cru}}_{f,i,j}$ or $v^{\text{cru}}_{f,i,j} < v^{\text{in}}_{f,i,j} < v^{\text{turn}}_f$, constraint (26a) reduces to $a^{\text{turn}}_{f,i,j} \leq a_{f,i,j}$. Since $a_{f,i,j} \leq a^{\text{turn}}_{f,i,j}$ is required in (8), we can further obtain $a^{\text{turn}}_{f,i,j} = a_{f,i,j}$, i.e., let the time point that train f reaches the speed v^{turn}_f on cell (i,j) equals the arrival time of the train. The formulations similar to (25) and (26) can also be constructed for the outgoing phase.

The optimization problem including the objective function (1) and constraints (2)-(26), is called the P_{NLP} problem, among which there are if-then constraints, i.e., (12) and (15), and nonlinear constraints, i.e., (14) and (16).

3.2.2. Formulation of the P_{PWA} Model: the P_{NLP} Model approximated by using PWA functions

This section proposes the MILP problem (P_{PWA}) by reformulating and approximating the nonlinear terms in the P_{NLP} problem, i.e., (12), (14), (15), and (16). A PWA function is adopted for the approximation, as well as three transformation properties proposed in Williams (2013), which are briefly introduced as below. Interested readers may refer to this reference for more details.

Let us consider the statement $\tilde{f}(\tilde{x}) \leq 0$, where $\tilde{f} : \mathbb{R}^n \to \mathbb{R}$ is affine, $\tilde{x} \in \chi$ with $\chi \subset \mathbb{R}^n$ and let $\tilde{Q} = \max_{\tilde{x}\in\chi} \tilde{f}(\tilde{x})$, $\tilde{q} = \min_{\tilde{x}\in\chi} \tilde{f}(\tilde{x})$.

- ***Transformation property I***: if we introduce a logical variable $l \in \{0,1\}$, then the following equivalence holds: $\left[\tilde{f}(\tilde{x}) \leq 0\right] \Leftrightarrow [l = 1]$ is true iff $\tilde{f}(\tilde{x}) \leq \tilde{Q} \cdot (1-l)$ and $\tilde{f}(\tilde{x}) \geq \varepsilon + (\tilde{q} - \varepsilon) \cdot l$.
- ***Transformation property II***: the product of two logical variables l_1 and l_2 can be replaced by an auxiliary logical variable $l_3 = l_1 \cdot l_2$, i.e., $[l_3 = 1] \Leftrightarrow [l_1 = l_2 = 1]$, which is equivalent to three linear inequalities: $-l_1 + l_3 \leq 0$, $-l_2 + l_3 \leq 0$ and $l_1 + l_2 - l_3 \leq 1$.
- ***Transformation property III***: the product $l \cdot \tilde{f}(\tilde{x})$ can be replaced by the auxiliary real variable $r = l \cdot \tilde{f}(\tilde{x})$, which satisfies $[l = 0] \Rightarrow [r = 0]$ and $[l = 1] \Rightarrow \left[r = \tilde{f}(\tilde{x})\right]$. Then $r = l \cdot \tilde{f}(\tilde{x})$ is equivalent to four inequalities: $r \leq \tilde{Q} \cdot l$, $r \geq \tilde{q} \cdot l$, $r \leq \tilde{f}(\tilde{x}) - \tilde{q} \cdot (1-l)$ and $r \geq \tilde{f}(\tilde{x}) - \tilde{Q} \cdot (1-l)$.

Note that *Transformation property I* has been used to formulate (24) for the speed indicators in Table 3 of Section 3.2.1. Moreover, the if-then constraints (12) and (15) can be reformulated as linear constraints by using *Transformation property I* (for the sake of compactness, we do not present the details here).

To approximate the nonlinear terms, constraint (14a) for calculating $L^{\text{in}}_{f,i,j}$ is first reformulated as the following set of linear constraints by using the logical speed indicators $\zeta_{1,f,i,j}$, $\zeta_{3,f,i,j}$, and $\zeta_{4,f,i,j}$:

$$-\tfrac{(v^{\text{cru}}_{f,i,j})^2-(v^{\text{in}}_{f,i,j})^2}{2\cdot\beta_{f,i,j}} - M\cdot\zeta_{1,f,i,j} \leq L^{\text{in}}_{f,i,j} \leq -\tfrac{(v^{\text{cru}}_{f,i,j})^2-(v^{\text{in}}_{f,i,j})^2}{2\cdot\beta_{f,i,j}} + M\cdot\zeta_{1,f,i,j}, \tag{27a}$$

$$\tfrac{(v^{\text{cru}}_{f,i,j})^2-(v^{\text{in}}_{f,i,j})^2}{2\cdot\alpha_{2,f,i,j}} - M\cdot(2-\zeta_{1,f,i,j}-\zeta_{3,f,i,j}) \leq L^{\text{in}}_{f,i,j} \leq \tfrac{(v^{\text{cru}}_{f,i,j})^2-(v^{\text{in}}_{f,i,j})^2}{2\cdot\alpha_{2,f,i,j}} + M\cdot(2-\zeta_{1,f,i,j}-\zeta_{3,f,i,j}), \tag{27b}$$

$$\tfrac{(v^{\text{cru}}_{f,i,j})^2-(v^{\text{in}}_{f,i,j})^2}{2\cdot\alpha_{1,f,i,j}} - M\cdot(2-\zeta_{1,f,i,j}-\zeta_{4,f,i,j}) \leq L^{\text{in}}_{f,i,j} \leq \tfrac{(v^{\text{cru}}_{f,i,j})^2-(v^{\text{in}}_{f,i,j})^2}{2\cdot\alpha_{1,f,i,j}} + M\cdot(2-\zeta_{1,f,i,j}-\zeta_{4,f,i,j}), \tag{27c}$$

$$\begin{aligned}\tfrac{(v^{\text{turn}}_f)^2-(v^{\text{in}}_{f,i,j})^2}{2\cdot\alpha_{1,f,i,j}} + \tfrac{(v^{\text{cru}}_{f,i,j})^2-(v^{\text{turn}}_f)^2}{2\cdot\alpha_{2,f,i,j}} - M\cdot(1-\zeta_{1,f,i,j}+2\cdot\zeta_{3,f,i,j}+2\cdot\zeta_{4,f,i,j}) \leq L^{\text{in}}_{f,i,j}\\ \leq \tfrac{(v^{\text{turn}}_f)^2-(v^{\text{in}}_{f,i,j})^2}{2\cdot\alpha_{1,f}} + \tfrac{(v^{\text{cru}}_{f,i,j})^2-(v^{\text{turn}}_f)^2}{2\cdot\alpha_{2,f,i,j}} + M\cdot(1-\zeta_{1,f,i,j}+2\cdot\zeta_{3,f,i,j}+2\cdot\zeta_{4,f,i,j})\,.\end{aligned} \tag{27d}$$

These constraints satisfy the uniformly accelerating and decelerating motions, and the detailed explanation of (27) is provided in Appendix A.3. Constraints similar to (27) can also be constructed for reformulating (14c) and for further calculating $L^{\text{out}}_{f,i,j}$, but for the sake of compactness, we do not report those details here. Let $\varpi^{\text{in}}_{f,i,j}$, $\varpi^{\text{cru}}_{f,i,j}$, and $\varpi^{\text{out}}_{f,i,j}$ be the square of $v^{\text{in}}_{f,i,j}$, $v^{\text{cru}}_{f,i,j}$, and $v^{\text{out}}_{f,i,j}$ respectively, as formulated in (28):

$$\varpi^{\text{in}}_{f,i,j} = \left(v^{\text{in}}_{f,i,j}\right)^2, \quad \forall f \in F, (i,j) \in E_f, \tag{28a}$$

$$\varpi^{\text{cru}}_{f,i,j} = \left(v^{\text{cru}}_{f,i,j}\right)^2, \quad \forall f \in F, (i,j) \in E_f, \tag{28b}$$

$$\varpi^{\text{out}}_{f,i,j} = \left(v^{\text{out}}_{f,i,j}\right)^2, \quad \forall f \in F, (i,j) \in E_f. \tag{28c}$$

As a result, (27a)-(27d) become linear, and instead (28a)-(28c) are nonlinear and should be approximated. The reason that we first reformulate (14a) and (14c) as above is to reduce the number of nonlinear terms that need to be approximated, i.e., by introducing (28), (27) and those constraints for reformulating (14c) become linear. Regarding (14b) that calculates $L^{\text{cru}}_{f,i,j}$ for the cruising phase, an additional step is needed to reformulate the nonlinear term $x \cdot y$ as $\frac{(x+y)^2-(x-y)^2}{4}$, i.e., reformulating (14b) as follows:

$$L^{\text{cru}}_{i,j} = \frac{1}{4} \cdot \left[\left(v^{\text{cru}}_{f,i,j} + d^{\text{cru}}_{f,i,j} - a^{\text{cru}}_{f,i,j}\right)^2 - \left(v^{\text{cru}}_{f,i,j} - d^{\text{cru}}_{f,i,j} + a^{\text{cru}}_{f,i,j}\right)^2\right]. \tag{29}$$

Then, by defining

$$m_{f,i,j} = \left(v^{\text{cru}}_{f,i,j} + d^{\text{cru}}_{f,i,j} - a^{\text{cru}}_{f,i,j}\right)^2, \tag{30a}$$

$$n_{f,i,j} = \left(v^{\text{cru}}_{f,i,j} - d^{\text{cru}}_{f,i,j} + a^{\text{cru}}_{f,i,j}\right)^2, \tag{30b}$$

equation (29) becomes linear, and instead (30a)-(30b) need to be approximated by using PWA functions, as will be explained next.

Based on the above reformulation, the nonlinear constraints (16), (28), and (30) need to be further approximated by using PWA functions. For simplicity, we only describe the approximating process of (28a) here; a similar process can be followed for approximating the other nonlinear constraints.

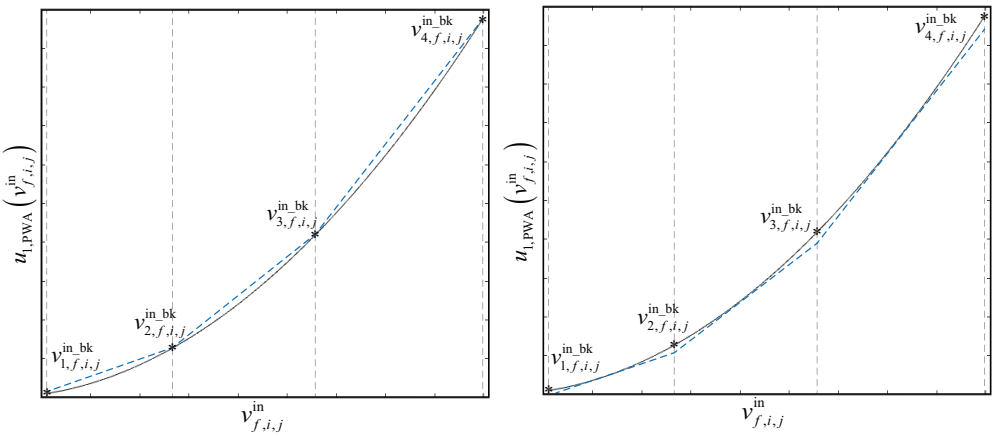


Fig. 3. The PWA approximation of the non-linear function

We adopt an approximation using three affine sub-functions as illustrated in Fig. 3. Note that more affine sub-functions can be selected if needed; the approach then stays similar in such a case. We consider two kinds of line fitting methods, namely the upper/lower line fitting method, where the values of the approximated line segments are no less/greater than the original curve, as shown in Fig. 3(a)-Fig. 3(b) respectively. The relevant coefficients regarding the three line segments (e.g., $v^{\text{in\_bk}}_{2,f,i,j}$ and $v^{\text{in\_bk}}_{3,f,i,j}$) are determined through minimizing the approximation errors between the original curve (indicated in black) and three line segments (indicated in blue). It is worth noting that the reason of using these two methods is to keep the approximated

constraints feasible. For instance, constraint (28a) should be approximated by using the lower line fitting method in Fig. 3(b), in order to guarantee that the approximated value of the train speed is not greater than its actual value and the corresponding speed limitation as well. Additionally, the approximated value of the time, the distance, and the square of the train speed should not be negative, so we keep all approximated values non-negative.

The PWA approximation of the nonlinear function (28a) over the interval $\left[\min\left(v^{\text{in}}_{f,i,j}\right), \max\left(v^{\text{in}}_{f,i,j}\right)\right]$, i.e., $\left[v^{\text{in\_bk}}_{1,f,i,j}, v^{\text{in\_bk}}_{4,f,i,j}\right]$, can be written as

$$u_{1,\text{PWA}}\left(v^{\text{in}}_{f,i,j}\right) = \varpi^{\text{in}}_{f,i,j} = \begin{cases} \mu_{1,f,i,j} \cdot v^{\text{in}}_{f,i,j} + \eta_{1,f,i,j}, \text{ if } v^{\text{in\_bk}}_{1,f,i,j} \le v^{\text{in}}_{f,i,j} \le v^{\text{in\_bk}}_{2,f,i,j} \\ \mu_{2,f,i,j} \cdot v^{\text{in}}_{f,i,j} + \eta_{2,f,i,j}, \text{ if } v^{\text{in\_bk}}_{2,f,i,j} \le v^{\text{in}}_{f,i,j} \le v^{\text{in\_bk}}_{3,f,i,j} \\ \mu_{3,f,i,j} \cdot v^{\text{in}}_{f,i,j} + \eta_{3,f,i,j}, \text{ if } v^{\text{in\_bk}}_{3,f,i,j} \le v^{\text{in}}_{f,i,j} \le v^{\text{in\_bk}}_{4,f,i,j} \end{cases} \tag{31}$$

where $\mu_{x,f,i,j}$ and $\eta_{x,f,i,j}$ are coefficients, $x = 1, ..., 3$.

Let us consider the logical variables $\lambda_{1,f,i,j}$ and $\lambda_{2,f,i,j}$ to satisfy the conditions $\left[v^{\text{in}}_{f,i,j} - v^{\text{in\_bk}}_{2,f,i,j} \le 0\right] \Leftrightarrow [\lambda_{1,f,i,j} = 1]$ and $\left[v^{\text{in}}_{f,i,j} - v^{\text{in\_bk}}_{3,f,i,j} \le 0\right] \Leftrightarrow [\lambda_{2,f,i,j} = 1]$, which can be represented as a set of linear inequalities by using Transformation property I (Williams 2013). Then, the function (31) can be rewritten as

$$\begin{aligned} u_{1,\text{PWA}}\left(v^{\text{in}}_{f,i,j}\right) = \varpi^{\text{in}}_{f,i,j} = {} & \lambda_{1,f,i,j} \cdot \lambda_{2,f,i,j} \cdot \left(\mu_{1,f,i,j} \cdot v^{\text{in}}_{f,i,j} + \eta_{1,f,i,j}\right) \\ & + (1 - \lambda_{1,f,i,j}) \cdot \lambda_{2,f,i,j} \cdot \left(\mu_{2,f,i,j} \cdot v^{\text{in}}_{f,i,j} + \eta_{2,f,i,j}\right) \\ & + (1 - \lambda_{1,f,i,j}) \cdot (1 - \lambda_{2,f,i,j}) \cdot \left(\mu_{3,f,i,j} \cdot v^{\text{in}}_{f,i,j} + \eta_{3,f,i,j}\right) \end{aligned} \tag{32}$$

We introduce the auxiliary logical variable $\lambda_{3,f,i,j}$ to replace the product $\lambda_{1,f,i,j} \cdot \lambda_{2,f,i,j}$. According to Transformation property II, the condition $\lambda_{3,f,i,j} = \lambda_{1,f,i,j} \cdot \lambda_{2,f,i,j}$ can also be rewritten as a system of linear inequalities. Moreover, by defining new auxiliary variables $z_{x,f,i,j} = \lambda_{x,f,i,j} \cdot v^{\text{in}}_{f,i,j}$, $x = 1, ..., 3$, which can be expressed as a set of linear inequalities by adapting Transformation property III, the function (32) can be further rewritten as

$$\begin{aligned} u_{1,\text{PWA}}\left(v^{\text{in}}_{f,i,j}\right) = \varpi^{\text{in}}_{f,i,j} = {} & z_{3,f,i,j} \cdot (\mu_{1,f,i,j} - \mu_{2,f,i,j} + \mu_{3,f,i,j}) + z_{2,f,i,j} \cdot (\mu_{2,f,i,j} - \mu_{3,f,i,j}) \\ & + \lambda_{3,f,i,j} \cdot (\eta_{1,f,i,j} - \eta_{2,f,i,j} + \eta_{3,f,i,j}) + \lambda_{2,f,i,j} \cdot (\eta_{2,f,i,j} - \eta_{3,f,i,j}) \\ & - z_{1,f,i,j} \cdot \mu_{3,f,i,j} - \lambda_{1,f,i,j} \cdot \eta_{3,f,i,j} + \mu_{3,f,i,j} \cdot v^{\text{in}}_{f,i,j} + \eta_{3,f,i,j} \end{aligned} \tag{33}$$

Finally, the nonlinear constraints (28a) can be replaced by the linear equation (33) and those linear inequalities obtained by using the three transformation properties, three logical variables $\lambda_{1,f,i,j}, \lambda_{2,f,i,j}, \lambda_{3,f,i,j}$, and three auxiliary variables $z_{1,f,i,j}, z_{2,f,i,j}, z_{3,f,i,j}$. Similar process can be followed for approximating the nonlinear constraints (16), (28b), (28c), and (30) by applying the three transformation properties and introducing extra logical variables and auxiliary variables, thus we do not report those details in this paper.

In particular, the clearing time constraint (16) is approximated by using a piece-wise constant function. We can also use the transformation properties in Williams (2013) to approximate (16), similar to the approximating process of (28a).

The optimization problem including the objective function (1), constraints (2)-(11), (13), (17)-(26), (27), (29), (33), and those constraints for reformulating (12) and (15) and for approximating (16), (28b), (28c), and (30), which are not detailed in this paper, is called the P_{PWA} problem.

3.2.3. Formulation of the P_{TSPO} problem: considering multiple train speed profile options (TSPOs) generated in a preprocessing step

In this section, another MILP problem (P_{TSPO}) considering multiple TSPOs is developed. A preprocessing step is used to generate multiple TSPOs, in order to restrict the search only to an efficient subset of all possible TSPOs. We still refer to the notations in Table 2, with the changes listed in Table 4. Note that the pre-generated TSPOs respect the formulas of the uniformly accelerating/decelerating motion and the technical requirements of train operations and infrastructures, e.g., train speed limitation, train acceleration/deceleration (which depends on traction/braking force), and length of block section.

Table 4. Changes of sets, subscripts, parameters, and variables for the $\mathrm{P}_{\mathrm{TSPO}}$ problem, compared with Table 2

Type of changes	Symbol	Description
added set	$Y_{f,i,j}$	the set of options of train speed profile vectors that train f may follow on cell (i,j), $\lvert Y_{f,i,j}\rvert$ is the number of TSPOs for train f on cell (i,j)
added subscript	b	TSPO index, $b_{f,i,j}=1,...,\lvert Y_{f,i,j}\rvert$, which indicates the TSPO index of train f on cell (i,j)
added parameter	$y_{f,i,j,b}$	the b^{th} train speed profile vector, $y_{f,i,j,b}\in Y_{f,i,j}$
added variable	$\vartheta_{f,i,j,b}$	binary variables, $\vartheta_{f,i,j,b}=1$ if the corresponding train speed vector $y_{f,i,j,b}$ is used by train f on cell (i,j), and otherwise $\vartheta_{f,i,j,b}=0$
added parameter	$y^{\mathrm{in}}_{f,i,j,b}, y^{\mathrm{cru}}_{f,i,j,b}, y^{\mathrm{out}}_{f,i,j,b}$	the b^{th} incoming, cruising, and outgoing speed of train f on cell (i,j), $y_{f,i,j,b}=\left[y^{\mathrm{in}}_{f,i,j,b}\quad y^{\mathrm{cru}}_{f,i,j,b}\quad y^{\mathrm{out}}_{f,i,j,b}\right]^{\top}\in Y_{f,i,j}$
changed to parameters	$L^{\mathrm{in}}_{f,i,j,b}/L^{\mathrm{out}}_{f,i,j,b}$	distance that train f runs over on cell (i,j) in the incoming/outgoing phase in the b^{th} train speed profile vector $y_{f,i,j,b}$
changed to parameters	$\zeta_{1,f,i,j,b},...,\zeta_{6,f,i,j,b}$	logical parameters to indicate the relation of the incoming, cruising, outgoing speed, and switching speed v^{turn}_{f} in the b^{th} train speed profile vector $y_{f,i,j,b}$, refer to Table 3

For each train on each cell, some train speed profile vectors $y_{f,i,j,b}$ are given, and each vector contains a possible set of incoming, cruising, and outgoing speeds, i.e., $y_{f,i,j,b}=\left[y^{\mathrm{in}}_{f,i,j,b}\quad y^{\mathrm{cru}}_{f,i,j,b}\quad y^{\mathrm{out}}_{f,i,j,b}\right]^{\top}$. Logical parameters $\zeta_{1,f,i,j,b},...,\zeta_{6,f,i,j,b}$ are used to indicate the speed conditions in the corresponding train speed profile vector $y_{f,i,j,b}$, as explained in Table 3. The problem objective is also to minimize the total train delay times at all visited stations, as formulated in (1). In addition, some constraints used by the $\mathrm{P}_{\mathrm{TSPO}}$ problem are presented as follows:

$$v^{\mathrm{in}}_{f,i,j}=\sum_{b=1}^{|Y_{f,i,j}|}\vartheta_{f,i,j,b}\cdot y^{\mathrm{in}}_{f,i,j,b},\quad \forall f\in F,(i,j)\in E_f, \tag{34}$$

$$v^{\mathrm{cru}}_{f,i,j}=\sum_{b=1}^{|Y_{f,i,j}|}\vartheta_{f,i,j,b}\cdot y^{\mathrm{cru}}_{f,i,j,b},\quad \forall f\in F,(i,j)\in E_f, \tag{35}$$

$$v^{\mathrm{out}}_{f,i,j}=\sum_{b=1}^{|Y_{f,i,j}|}\vartheta_{f,i,j,b}\cdot y^{\mathrm{out}}_{f,i,j,b},\quad \forall f\in F,(i,j)\in E_f, \tag{36}$$

$$\sum_{b=1}^{|Y_{f,i,j}|}\vartheta_{f,i,j,b}=1,\quad \forall f\in F,(i,j)\in E_f \tag{37}$$

$$\vartheta_{f,i,j,b}\cdot\frac{\left(L^{\mathrm{cell}}_{i,j}-L^{\mathrm{in}}_{f,i,j,b}-L^{\mathrm{out}}_{f,i,j,b}\right)}{y^{\mathrm{cru}}_{f,i,j,b}}\le d^{\mathrm{cru}}_{f,i,j}-a^{\mathrm{cru}}_{f,i,j}\le\frac{\left(L^{\mathrm{cell}}_{i,j}-L^{\mathrm{in}}_{f,i,j,b}-L^{\mathrm{out}}_{f,i,j,b}\right)}{y^{\mathrm{cru}}_{f,i,j,b}}+M\cdot(1-\vartheta_{f,i,j,b}) \tag{38}$$

$$\frac{(d_{f,i,j}-a_{f,i,j})\cdot y^{\mathrm{out}}_{f,i,j,b}}{\varepsilon+y^{\mathrm{out}}_{f,i,j,b}}-M\cdot(1-\vartheta_{f,i,j,b})\le\tau^{\mathrm{approach}}_{f,j,k}\le\frac{(d_{f,i,j}-a_{f,i,j})\cdot y^{\mathrm{out}}_{f,i,j,b}}{\varepsilon+y^{\mathrm{out}}_{f,i,j,b}}+M\cdot(1-\vartheta_{f,i,j,b}),\quad \forall f\in F,(i,j)\in E_f,(j,k)\in E_f,b=1,...,|Y_{f,i,j}| \tag{39}$$

$$\tau^{\mathrm{clear}}_{f,p,i}=\sum_{b=1}^{|Y_{f,i,j}|}\frac{2\cdot L^{\mathrm{train}}_{f}\cdot\vartheta_{f,i,j,b}}{y^{\mathrm{in}}_{f,i,j,b}+y^{\mathrm{cru}}_{f,i,j,b}},\quad \forall f\in F,(p,i)\in E_f,(i,j)\in E_f \tag{40}$$

Constraints (34)-(36) determine the selected incoming, cruising, and outgoing speed respectively, i.e., if $\vartheta_{f,i,j,b}=1$, then $v^{\mathrm{in}}_{f,i,j}=y^{\mathrm{in}}_{f,i,j,b}$, $v^{\mathrm{cru}}_{f,i,j}=y^{\mathrm{cru}}_{f,i,j,b}$, and $v^{\mathrm{out}}_{f,i,j}=y^{\mathrm{out}}_{f,i,j,b}$. Constraint (37) ensures that one and only one TSPO is selected for each train on each cell. Constraint (38) is the cell length constraint, which restricts the distance that a train runs over on a cell. Specifically, if $\vartheta_{f,i,j,b}=1$, i.e., the b^{th} train speed profile vector $y_{f,i,j,b}$ is used, constraint (38) reduces to a linear equation $d^{\mathrm{cru}}_{f,i,j}-a^{\mathrm{cru}}_{f,i,j}=\frac{\left(L^{\mathrm{cell}}_{i,j}-L^{\mathrm{in}}_{f,i,j,b}-L^{\mathrm{out}}_{f,i,j,b}\right)}{y^{\mathrm{cru}}_{f,i,j,b}}$, which satisfies the basic formula "time $=\frac{\text{distance}}{\text{constant speed}}$" of the uniform motion. Constraints (39) and (40) define

the approach time and clearing time respectively. Note that if train f stops on cell (i,j), i.e., $\vartheta_{f,i,j,b} = 1$ and $y^{\text{out}}_{f,i,j,b} = 0$, the approach time of train f on the successive cell (j,k) should be zero. To avoid the error that the denominator is zero, a sufficiently small positive number ε is used in (39).

The optimization problem including the objective function (1), constraints (2)-(4), (8)-(11), (17)-(23), (25)-(26), and (34)-(40), is called the P_{TSPO} problem.

4. Solution approaches

In this section, we introduce the solution approaches for solving the proposed optimization approaches, i.e., a two-level approach for solving the P_{NLP} problem and a custom-designed two-step approach for solving the P_{TSPO} problem. Regarding the solution approach of the P_{PWA} problem, an MILP solver can be used, such as CPLEX or Gurobi.

4.1. A two-level approach for solving the P_{NLP} *problem*

The nonlinear dynamics of the P_{NLP} problem limit its scalability and applicability for large-scale instances. Thus, we propose a two-level approach to solve the P_{NLP} problem, as illustrated in Fig. 4(a), where a genetic algorithm based heuristic is introduced to generate the possible train orders based on the track layouts, train routes, delays, etc. in the upper level, and a nonlinear programming method is used in the lower level to optimize the departure/arrival times and the train speed profiles under the fixed train orders.

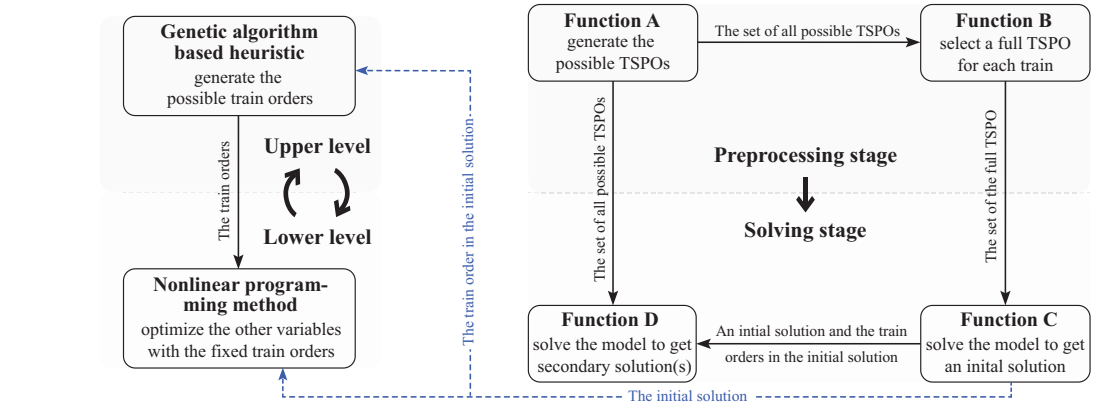
In the upper level, to describe the entire set of train orders in the network, we use a chromosome. This is defined as a vector that is composed by several sub-vectors. There is a sub-vector for each merging/diverging point (i.e., where train orders can change; we call them relevant points in what follows) of the network. A sub-vector is used to indicate the train orders at that specific relevant point. In order to generate feasible initial populations, the train orders defined in the original train timetable or the initial solution can be used as a starting point. In addition, we only adopt the mutation operation for the genetic algorithm used in this paper to generate feasible chromosomes. In particular, the mutation operation is carried out by swapping the order of two trains at a relevant point inside the chromosomes. Since the orders of trains at the relevant points are related to each other, the order of these two chosen trains at other relevant points may need to be swapped accordingly. Furthermore, the train delays at the relevant points are also used as a supplement for the decision of swapping trains. After a new population is generated, the nonlinear programming method in the lower level is used to optimize the departure/arrival times and train speed profiles and to obtain the fitness for each chromosome. We terminate the genetic algorithm after a given number of generations, i.e., 10 generations considered in our case.

Due to the non-convexity of using the P_{NLP} problem for the nonlinear optimization problem, the two-level approach can only obtain a local minimum for the departure/arrival times and speeds, by given the train orders; therefore, the final solution of the nonlinear optimization problem is a local minimum associated with the best upper level solution. The two-level approach with multiple initial solutions (including multiple initial train orders for the upper level and multiple initial departure/arrival times and train speeds for the lower level) could improve the performance, but reaching the global optimum can in general not be guaranteed. The initial solution could be the original timetable or the initial solution obtained by the P_{TSPO} problem through considering a fixed full TSPO for each train, as indicated by the blue dashed line in Fig. 4.

4.2. A custom-designed two-step approach for solving the P_{TSPO} *problem*

The P_{TSPO} problem is an MILP problem that can be solved by a standard MILP solver. Inspired by the good performance on similar problems in Xu et al. (2017), a custom designed two-step approach is particularly developed to solve the P_{TSPO} problem, in order to speed up the solving procedure, as illustrated in Fig. 4(b).

As the P_{TSPO} problem is defined by considering multiple pre-determined TSPOs, a preprocessing stage is used to generate the possible TSPOs (by Function A) and to clarify the full TSPO (by Function B). Each TSPO generated by Function A respects the formulas of the uniformly accelerating/decelerating motion and the technical requirements of train operations and infrastructures, e.g., train speed limitation, train



(a) The two-level approach for solving the P_{NLP} model (b) The custom-designed two-step approach for solving the P_{TSPO} model

Fig. 4. Illustration of the solution approaches

acceleration/deceleration (which depends on traction/braking force), and length of block section. The full TSPO for each train derives from the corresponding set of all possible TSPOs, by selecting the fastest TSPO from this set that lets the train run as fast as possible. In Function C of the solving stage, we consider the selected full TSPO only to solve the P_{TSPO} problem by using a standard MILP solver, which results in an initial solution (i.e., an upper bound with a fixed full TSPO for each train). Then, the obtained initial solution is given as one feasible solution to the MILP solver, for solving the P_{TSPO} problem with the larger set of all possible TSPOs. Therefore, in Function D, an improved secondary solution can be obtained through optimizing the TSPOs (and optimizing the train orders as well). Moreover, the train orders of the initial solution can also be given as an input of the problem in Function D; as a result, we can obtain an improved secondary solution with fixed train orders. Due to the limited number of TSPOs resulting from the preprocessing stage, only a local optimal solution can be obtained for the P_{TSPO} problem and its performance strongly depends on the given subset of TSPOs.

5. Numerical experiments

Before reporting the experimental results, we first describe the dataset in Section 5.1, i.e., a Dutch railway network. In Section 5.2, we compare the overall performance of the three proposed optimization approaches based on the Dutch test case described in Section 5.1. For the P_{PWA} problem and the P_{TSPO} problem, we have multiple computational configurations; therefore, we further investigate the impact of these configurations on the results. In Sections 5.3, the analysis of the P_{PWA} problem focuses on assessing the effectiveness of the approximation when using different line fitting methods, from the viewpoints of feasibility and approximation error. For the P_{TSPO} problem, Sections 5.4 investigates the impact of the TSPOs generated in the preprocessing step on the solution quality, by considering different sets of discrete speed values. Moreover, we explore the benefits of changing train orders and managing train speeds. Finally, a lower bound is generated to evaluate the quality of the P_{TSPO} solution obtained within a given computation time limit. Moreover, we additionally report the detailed data about the solutions of this test case in the online repository (Research Collection ETH Zurich). In Appendix C, we explore the applicability of the proposed approach to a different test case adapted from INFORMS RAS (2012), in order to show the generality of the conclusions.

We use the SNOPT solver implemented in the MATLAB (R2016a) TOMLAB toolbox to solve the MINLP problem, i.e., the P_{NLP} problem, by applying the two-level approach introduced in Section 4. We adopt the IBM ILOG CPLEX optimization studio 12.6.3 with default settings to solve the MILP problems, i.e., the P_{PWA} problem and the P_{TSPO} problem. The custom-designed two-step approach described in Section 4.2 is particularly considered for the P_{TSPO} problem. The experiments are all performed on a computer with an Intel® Core™ i7 @ 2.00 GHz processor and 16GB RAM.

5.1. Description of the experimental dataset

We consider the line of the Dutch railway network, connecting Utrecht (Ut) to Den Bosch (Ht), of about 50 km length. The network under consideration is shown in Fig. 5. The network is composed of 40 nodes and 42 cells, with 2 main tracks, divided into a long corridor for each traffic direction and 9 stations. The two tracks in different directions are independent, so only one direction is considered, i.e., from Utrecht (Ut) to Den Bosch (Ht). Three categories of trains are considered: intercity, sprinter, and freight trains, with different acceleration, deceleration, and dynamic characteristics. Four global[1] routes (identified by color: blue for intercity, green for sprinter, red for freight) are determined and graphically presented in the lower part of Fig. 5, in terms of origin, intermediate stop, destination, and number of trains per hour. We consider one hour of traffic based on a regular-interval timetable, with 15 trains. Sprinter trains stop at all stations; intercity and freight trains stop only at the origin and destination stations.

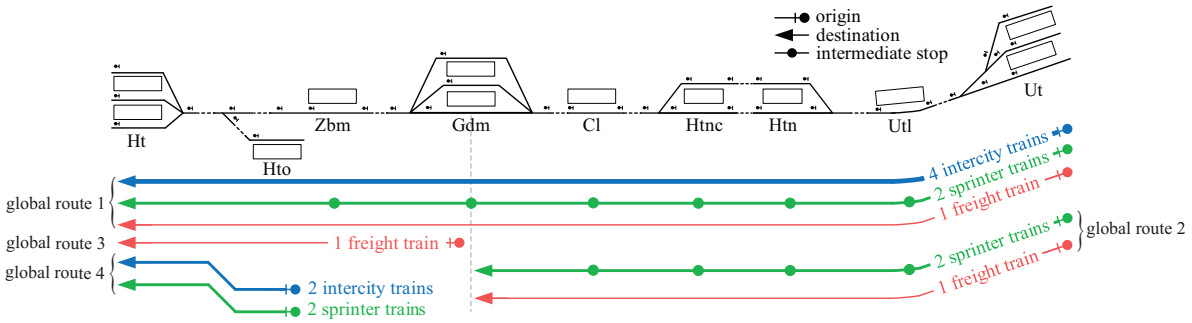


Fig. 5. A real-world experimental network adapted from the Dutch railway network

Each train is given a randomly generated primary delay time c_f^{pri} at its origin. More specifically, we consider 10 delay cases of the primary delays following a 3-parameter Weibull distribution. The delay distributions differ per train category, and the following parameters in the form of [scale, shape, shift] are used: 1) for intercity trains, [394, 2.27, 315]; 2) for sprinter trains, [235, 3.00, 186]; 3) for freight trains, [1099, 2.62, 885]. These values come from fitting to real-life data as explained in Corman et al. (2011b).

5.2. Performance evaluation of the P_{NLP} *problem, the* P_{PWA} *problem, and the* P_{TSPO} *problem*

In this section, we use the Dutch test case introduced in Section 5.1 to evaluate the overall performance of the three proposed optimization approaches, from the point of view of effectiveness and efficiency.

We assess the performance of the three proposed optimization approaches on multi-scale instances, i.e., considering several instances with different numbers of trains (ranging from 2 to 15, a subset of the 15 trains described in Fig. 5) and with heterogeneous traffic. We here consider two computation time limits (i.e., 180 and 3600 seconds) for all three proposed optimization problems, and we output the best feasible solution obtained within each given computation time limit. A large set of TSPOs (i.e., Set_1 in Table 5) is used here for the P_{TSPO} problem, due to its good solution quality, as will be discussed in Section 5.4. Moreover, we consider two scenarios for the P_{PWA} problem regarding the upper and lower line fitting methods used for approximating the nonlinear constraints, indicated as "PWA_ul" and "PWA_ll", as will be explained in Section 5.3.

In some experiments of the P_{PWA} problem, we cannot obtain any feasible solution within the given computation time limit; therefore, in Fig. 6, we particularly report the average results of the three proposed optimization approaches respectively for the corresponding feasible cases of the P_{PWA} problem. The bars indicate the total train delay time, and refer to the Y-axis on the left-hand side, and the lines (with symbols) indicate the actual computation time, and refer to the Y-axis on the right-hand side. A missing bar/line

[1] A global route identifies the origin and destination of a train service, but does not specify tracks and platforms used in station areas. The tracks and platforms used in a station area are described as local routes.

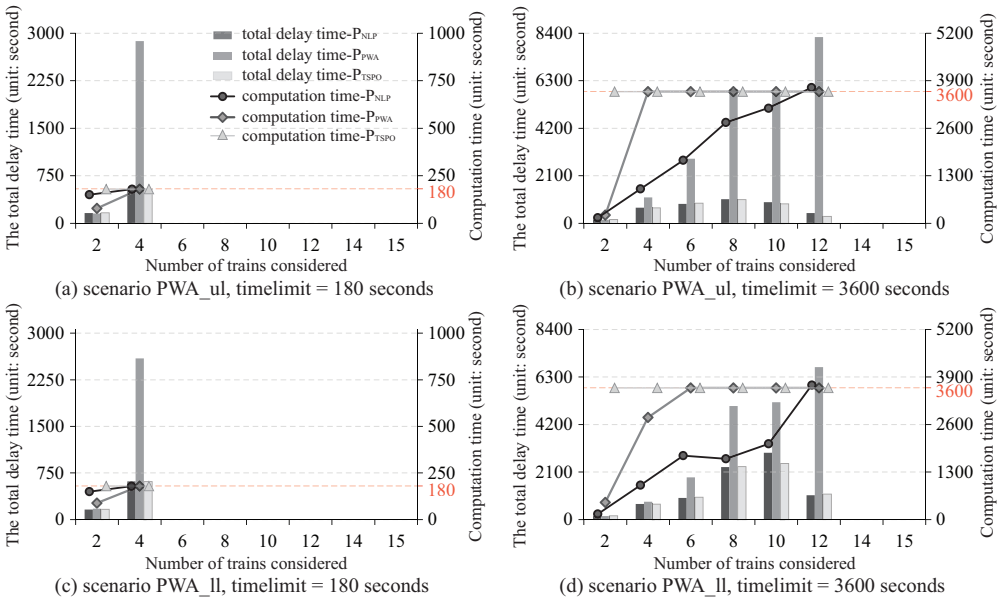


Fig. 6. Results of the three optimization approaches, corresponding to the feasible cases of the P_{PWA} approach

means that no feasible solution is found for the given instance. Fig. 6(a) and Fig. 6(b) correspond to the "PWA_ul" scenario of the P_{PWA} problem, and Fig. 6(c) and Fig. 6(d) correspond to the "PWA_ll" scenario. Fig. 6(a) and Fig. 6(c) illustrate the results obtained within 180 seconds of computation time, and Fig. 6(b) and Fig. 6(d) give the results obtained within 3600 seconds.

We can see that the solution quality of the P_{PWA} problem is the worst in most instances, as the dark gray bars are much higher than the other bars, even when the computation time is extended to 3600 seconds. The solution quality of the P_{NLP} problem and the P_{TSPO} problem is similar in most instances, with a deviation of less than 33% (corresponding to a delay time of 151 seconds). When focusing on the computational efficiency, the P_{NLP} problem appears to perform better on small-scale instances, because the black line (with dots) is mostly lower than the light gray line (with triangles) for the instances with less than 10 trains, as is shown in Fig. 6(b) and Fig. 6(d).

As the P_{NLP} problem and the P_{TSPO} problem can obtain feasible solutions for all delay cases, we next focus on all the results of the 10 delay cases to further evaluate the performance of these two optimization approaches, instead of only considering the corresponding feasible cases of the P_{PWA} problem. Fig. 7 comparatively presents the results of these two models, as an average of the 10 delay cases, in terms of the objective value (i.e., the total train delay time), the actual computation time, and the improvement in solution quality. Fig. 7(a) has the same structure as Fig. 6. In Fig. 7(b), each black (white) bar indicates the average improvement in solution quality for each instance, when comparing the P_{NLP} solution with the P_{TSPO} solution obtained within 180 (3600) seconds respectively, i.e., $\frac{P_{NLP}\text{ solution}-P_{TSPO}\text{ solution}}{P_{NLP}\text{ solution}} \times 100\%$. A positive value means that the solution quality of the P_{TSPO} problem is better, while a negative value implies a better solution quality of the P_{NLP} problem.

As illustrated in Fig. 7, the solution quality of the P_{NLP} problem and the P_{TSPO} problem differs among instances. Regarding the instances with a larger number of trains (i.e., 8-15 trains), much better solutions are found by the P_{TSPO} problem, attaining a 30% improvement in the solution quality at most. The P_{TSPO} solution found within 180 seconds is even better than the P_{NLP} solution obtained by consuming a longer computation time (which extends to 3600 seconds). In the other instances with smaller scales, the P_{NLP} problem performs better, as a solution with a smaller train delay time can be found. Although the P_{NLP} problem can find better solutions in small-scale instances, in comparison, the P_{TSPO} solution obtained within

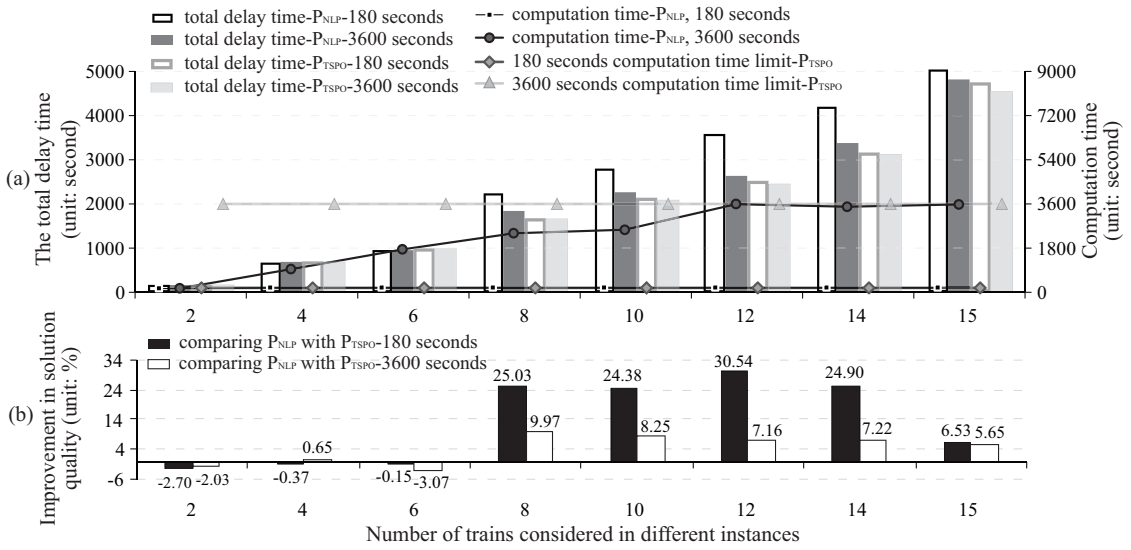


Fig. 7. Results of the P_{NLP} problem and the P_{TSPO} problem

180 seconds of computation time is still satisfactory. The P_{TSPO} solution is 3% worse at most than the P_{NLP} solution, which is relatively small when comparing to the 30% improvement of the P_{TSPO} problem achieved for larger-scale instances. Overall, the performance of the P_{TSPO} problem is the best, as a solution with a good quality can be found efficiently (within 180 seconds). Moreover, the train timetables (dispatching solutions) and the speed-space graphs obtained by the P_{NLP} problem and the P_{TSPO} problem for the Dutch test case are provided in Fig. 14 - Fig. 14(d) of Appendix B.

5.3. Further analysis of the experimental results of the P_{PWA} *problem*

We now study the solution quality and computational efficiency of the P_{PWA} problem by considering different line fitting methods (namely the upper and lower line fitting methods, as illustrated in Fig. 3), and we also analyze the resulting approximation errors. As discussed before, in order to guarantee the feasibility of the approximated constraints, we only use the lower line fitting method in Fig. 3(b) to approximate (28). Regarding the approximation of (30), we consider both the upper and lower line fitting methods, which results in two scenarios, indicated as "PWA_ul" and "PWA_ll" respectively, and we further explore the impact of the line fitting method on the solution quality. We also use the Dutch railway network in Fig. 5 as test bed, and we consider different instances with different numbers of trains (ranging from 2 to 15, a subset of the 15 trains described in Fig. 5) and with heterogeneous traffic.

The CPLEX solving process of the P_{PWA} problem is terminated by considering a given computation time limit (i.e., 180 seconds and 3600 seconds), and we then output the best feasible solution obtained within the given computation time limit. Fig. 8 illustrates the relevant results of "PWA_ul" and "PWA_ll" for each computation time limit, indicated as dark bars and light bars respectively. A missing bar means that no feasible solution is found for the instance within the given computation time limit. Fig. 8(a) gives the number of the obtained feasible solutions, out of the 10 delay cases. Fig. 8(b)-Fig. 8(c) present the actual computation time and the objective value as an average of the 10 delay cases.

The optimal solution can be obtained when considering only 2 trains (and 4 trains in "PWA_ll" scenario as well), as the actual computation time of these instances is less than the given computation time limit. For the other instances, the optimality cannot be guaranteed. A longer computation time leads to better objective values and a larger number of cases for which a feasible solution can be attained. No feasible solution can be obtained within 180 seconds for the instances with more than 4 trains, and no feasible solution is obtained within 3600 seconds for the instances with more than 12 trains. Moreover, "PWA_ll" yields a

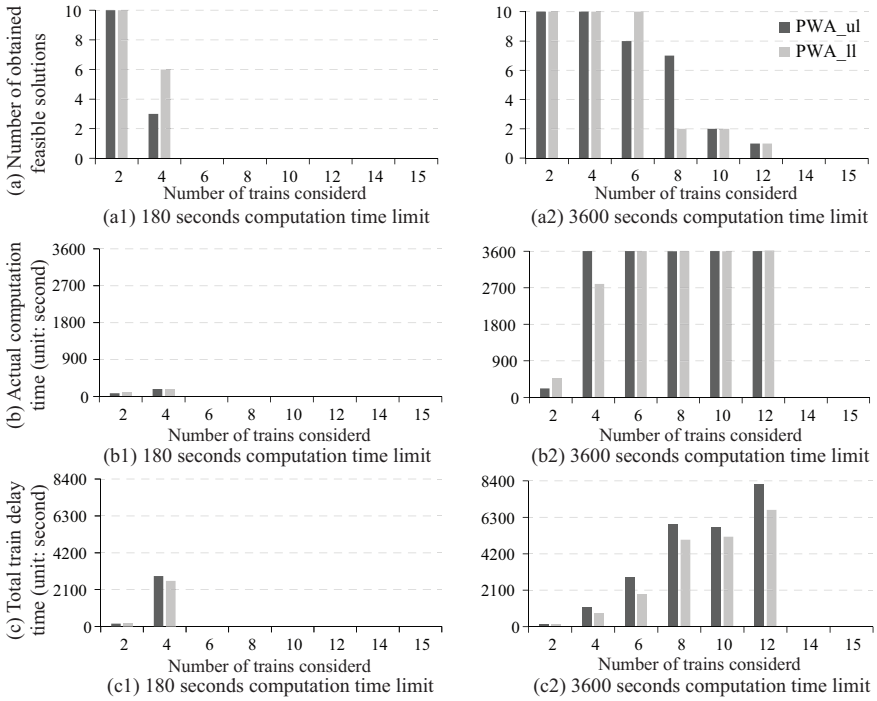


Fig. 8. Results of the $\mathrm{P_{PWA}}$ problem, for "PWA_ul" and "PWA_ll"

better performance in most instances, as it attains more feasible solutions, relatively shorter computation times, and smaller objective values.

The approximation errors of "PWA_ul" and "PWA_ll" for different constraints of the $\mathrm{P_{PWA}}$ problem are presented in Fig. 9, as the percentage, i.e., $\frac{|\text{approximated value}-\text{actual value}|}{\text{actual value}} \times 100\%$, and as an average of the 10 delay cases. The errors caused by approximating (30a) and (30b) lead to a deviation for calculating L^{cru} in (29), so we directly analyze the deviation value (approximation error) of L^{cru} in (29). The (blue) diamond, (green) square, (pink) dot, and (orange) triangle symbols indicate the approximation errors in the final solution for (29), (28a), (28b), and (28c) respectively. The dark small symbols indicate the approximation error of the solution obtained within 180 seconds of computation time, and the light large symbols represent the approximation error of the solution obtained within 3600 seconds. A missing symbol means that no feasible solution is found within the given computation time limit, i.e., the dark small symbols for the instances considering more than 4 trains and the light large symbols for the instances with 14-15 trains.

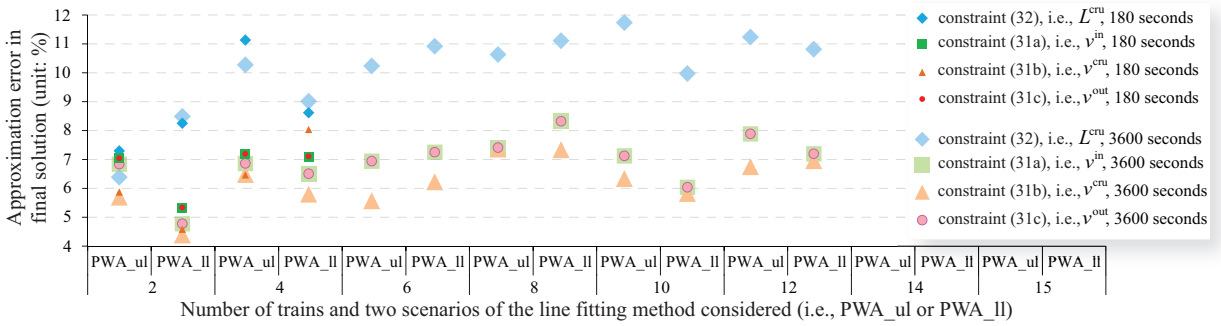


Fig. 9. Approximation errors of the constraints in the $\mathrm{P_{PWA}}$ problem, for "PWA_ul" and "PWA_ll"

As illustrated in Fig. 9, the performance of "PWA_ll" and "PWA_ul" differs among instances, i.e., "PWA_ll" performs better for the instances with 2, 4, 10, and 12 trains, while "PWA_ul" performs better for the instances with 6 and 8 trains. However, "PWA_ll" and "PWA_ul" overall perform similarly, with less than 2.5% difference of errors between them. Moreover, the approximation error of (29) is larger than that of the others, ranging from 6% to 12%, which results from the different magnitudes of the speed variable (v) and the time variables (a and d). The approximation error of (28b) is the smallest, and it ranges from 4% to 8%. For reducing the errors further, we can consider a PWA approximation using more affine subfunctions, and follow the approach described in Section 3.2.2.

Furthermore, we analyze the number of constraint violations caused by the PWA approximation. Regarding (28), no constraint is violated, as we apply the lower line fitting method to keep a smaller (positive) approximated value of the train speed than its actual value. For (29), around 5% (ranging from 4.2% to 5.0% for "PWA_ll" and from 4.1% to 5.6% for "PWA_ul") of the constraints is violated, in the sense that the approximated distance that a train travels in the cruising phase is larger than the actual distance that a train can move.

In summary, from all perspectives, i.e., the solution quality, the computational efficiency, the feasibility, and the errors, the $\mathrm{P_{PWA}}$ problem do not seem to perform good enough for addressing the integrated problem of traffic management and train control.

5.4. Further analysis of the experimental results of the $\mathrm{P_{TSPO}}$ *problem*

We now study the impact of the TSPOs generated in the preprocessing step on the solution quality. Six sets of TSPOs are generated by considering different discrete speed values for different train categories presented in Table 5, denoted as Set_1, ..., Set_6 respectively. Note that intercity and sprinter trains use the same speed pattern in each set. The number of the discrete speed values used in Set_1, ..., Set_6 is decreasing, which implies that the resolution of the train speed becomes lower and less TSPOs are available. The total number of TSPOs corresponding to the 6 sets is provided in columns 4-5 of Table 5. Column 4 gives the total number of TSPOs per train per block section, i.e., summing up the number of TSPOs for each train on each block section; column 5 presents the number of possibilities of combining the TSPOs for the train services, which indicates the scale of the feasible solution space.

Table 5. Six sets of TSPOs generated by using different discrete speed values

	Discrete speed values for intercity train and sprinter train (unit: km/h)	Discrete speed values for freight train (unit: km/h)	Total number of TSPOs per train per block section	Number of all possibilities of combining the TSPOs
Set_1	$\{0, 40, 60, 80, 90, 100, 110, 120, 130\}$	$\{0, 20, 30, 40, 50, 60, 70, 80\}$	16402	5.70×10^{50}
Set_2	$\{0, 40, 70, 90, 100, 110, 120, 130\}$	$\{0, 20, 40, 50, 60, 70, 80\}$	12370	5.28×10^{46}
Set_3	$\{0, 40, 70, 90, 110, 120, 130\}$	$\{0, 20, 40, 60, 70, 80\}$	9084	3.16×10^{43}
Set_4	$\{0, 40, 70, 100, 120, 130\}$	$\{0, 20, 50, 70, 80\}$	6332	5.56×10^{39}
Set_5	$\{0, 40, 100, 130\}$	$\{0, 40, 80\}$	2388	8.27×10^{28}
Set_6	$\{0, 40, 130\}$	$\{0, 40, 80\}$	1278	6.71×10^{19}

Fig. 10 illustrates the results of the 6 sets as a function of the computation time, in particular, the total train delay time on average of the 10 delay cases. Note that the CPLEX solving process is terminated by considering 8 computation time limits ranging from 180 to 3600 seconds, and the best feasible solution obtained within each given computation time limit is presented. The 6 sets are distinguished by colors: green, blue, purple, pink, orange and yellow for Set_1, ..., Set_6 respectively. For each set, the result with fixed train orders is drawn as a solid line and the result considering variable train orders is indicated by a dashed line. Each line presents an initial solution (represented by a star) and secondary solutions (indicated by dot and square symbols) as a function of computation time. Recall that the initial solution is obtained by considering a fixed full TSPO for each train on each block section and then improved to generate the secondary solutions by considering a larger set of multiple TSPOs.

We first focus on the results with fixed train orders, presented as solid lines in Fig. 10. The initial optimal solution considering a fixed full TSPO for each train on each block section (i.e., each train is required to run as fast as possible with respect to the safety, technical, and operational requirements) can be obtained

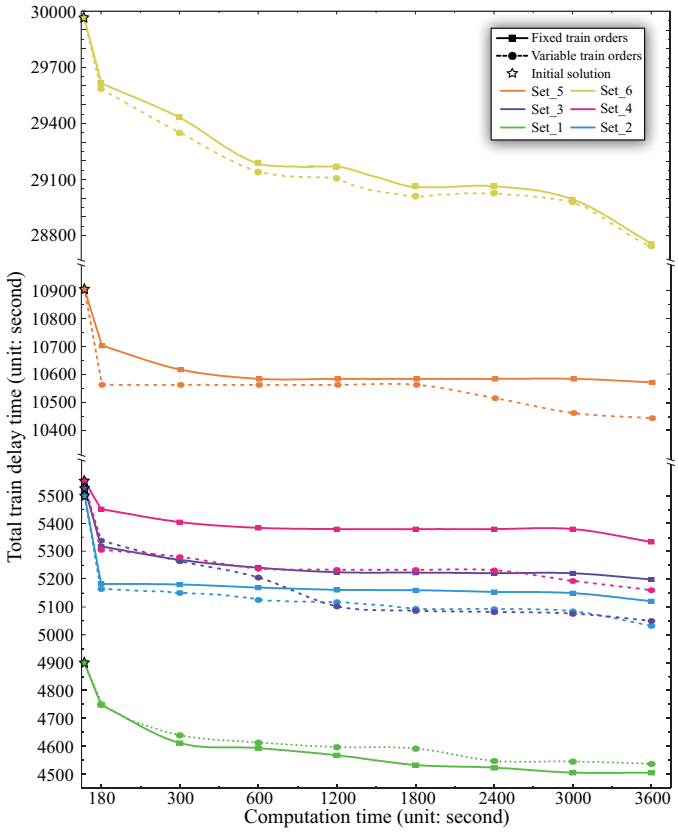


Fig. 10. Total train delay time of the 6 sets as a function of computation time

efficiently (i.e., less than 6 seconds). The initial solution is further improved to generate the secondary solutions by considering a larger set of multiple TSPOs. As shown, when focusing on one set, the total delay time decreases as a function of the computation time, implying an improvement in solution quality. *This demonstrates the benefit of integrating traffic management and train control, i.e., train delays can be reduced by managing train speed.* Moreover, focusing on all the 6 sets, the total delay time increases in both the initial solution and the secondary solutions, if fewer discrete speed values are considered. So the total delay time increases with a decreasing resolution of the train speed in Set_1, ..., Set_6 sequentially. This results from the reduced solution space, i.e., the reduced number of TSPOs available. The improvement in train delay time of Set_1 (the best/significant one with the lowest total delay time) is 3.14% at 180 seconds, and it increases to 8.08% when extending the computation time to 3600 seconds.

When comparing with the results with fixed train orders, the solution quality considering variable train orders is better for Set_2, ..., Set_6, i.e., the dashed line is mostly lower than the corresponding solid line. For Set_1, which contains the largest number of TSPOs among the 6 sets, the result considering variable train order is worse than that for fixed train orders. This may result from the large solution space caused by the huge number of TSPOs and various possibilities of train orders, and the high sensitivity of the solutions to the train speed. The sensitivity of the solutions to the train speed is higher with an increasing number of TSPOs. Therefore, the MILP solver is unable to explore the effective space (regarding train speed) within a given computation time limit. When reducing the solution space by fixing train orders, the MILP solver has a higher chance to explore the solution space more efficiently within the same time limit. To conclude, we may consider variable train orders for the case with a low resolution of the train speed, and fixed train orders for the case with a high train speed resolution, in order to obtain a better solution within a given computation time limit.

Fig. 11(a) and Fig. 11(b) present the percentage of improvement in solution quality from the initial solution as a function of computation time, for the cases considering fixed and variable train order respectively. This percentage of improvement is calculated by the formula $\frac{\Phi_{\ell-1}-\Phi_{\ell}}{\Phi_1-\Phi_9}$, for $\ell = 2, ..., 9$. Note that ℓ is the index of the computation time limits considered, i.e., $\ell = 1, ..., 9$ represent 0 (initial solution), 180, 300, 600, 1200, 1800, 2400, 3000, and 3600 seconds of computation time limits respectively; and Φ_ℓ indicates the total train delay time at the corresponding computation time limit ℓ. For instance, the delay time of the initial solution for Set_1 is 4902 seconds (i.e., $\Phi_1 = 4902$), which is reduced to 4748 and 4506 seconds in the secondary solutions obtained at 180 and 3600 seconds of computation time respectively (i.e., $\Phi_2 = 4748$ and $\Phi_9 = 4506$); the percentage of improvement in solution quality within 180 seconds is then $\frac{\Phi_1-\Phi_2}{\Phi_1-\Phi_9} = \frac{4902-4748}{4902-4506}$=39%. In each figure, the percentages of improvement in solution quality at the 8 computation time limits are respectively drawn from the left to the right using different colors, and each horizontal bar represents a set of TSPOs.

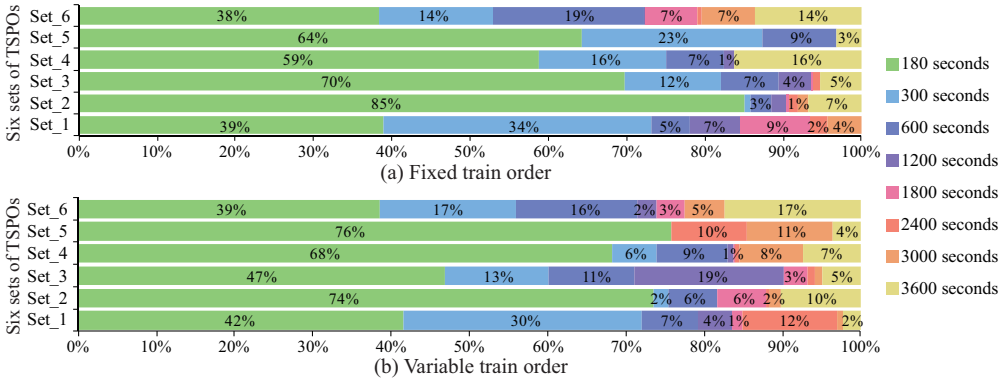


Fig. 11. Percentage of the improvement in solution quality as a function of computation time for the 6 sets

As illustrated, the green region (i.e., the improvement in solution quality at the first 180 seconds) occupies most of the space for each bar, ranging from 38% to 85% in Fig. 11(a) and from 39% to 76% in Fig. 11(b). When expanding the focus to the green and light blue portions, the percentage of the quality improvement from 0 to 300 seconds of computation time is more than a half for all the sets, achieving 52% - 87% in Fig. 11(a) and 56% - 76% in Fig. 11(b). This implies that a significant improvement in solution quality can be achieved efficiently. Although the solution quality can be improved by considering a longer computation time, the improvement is not as significant as that achieved within the first 180 seconds. Hence, practically, it is not a good choice to consume a much longer computation time for obtaining a small improvement only.

Although a significant improvement from the initial solution can be achieved efficiently, the solution quality is still unknown, i.e., how far is the solution away from the optimal one (an estimation of the optimality gap). Therefore, we generate lower bounds for the P_{TSPO} problem to assess their solution quality. The so-called lower bound here is not physically feasible and therefore not the best lower bound.

Fig. 12(a) and Fig. 12(b) illustrate the obtained lower bounds, feasible solutions, and the corresponding estimation of optimality gaps[1] as a function of the computation time, and as an average of 10 delay cases, considering fixed and variable train orders respectively. The largest set of TSPOs (i.e., Set_1) is used for computing the lower bounds, due to its good solution quality. The best feasible solutions obtained within the given computation time limits are represented by black dots (connected by a solid line), and the lower bound is indicated by a horizontal dashed line. The percentage in blue color indicates the optimality gap. To calculate these lower bounds, we have neglected train acceleration and deceleration characteristics, i.e., we assume that a train can suddenly and instantly accelerate or decelerate to any given speed value (listed in row 2 of Table 5). This leads to a reduction of the optimization problem to identify an optimal cruising

[1]Note that the gap between the feasible solution obtained and the lower bound is considered as an estimation of the optimality gap.

speed for each train on each block section, as the incoming speed and the outgoing speeds do not affect the final results anymore. The calculation of the lower bounds is also an MILP problem, so we use the CPLEX solver to get them.

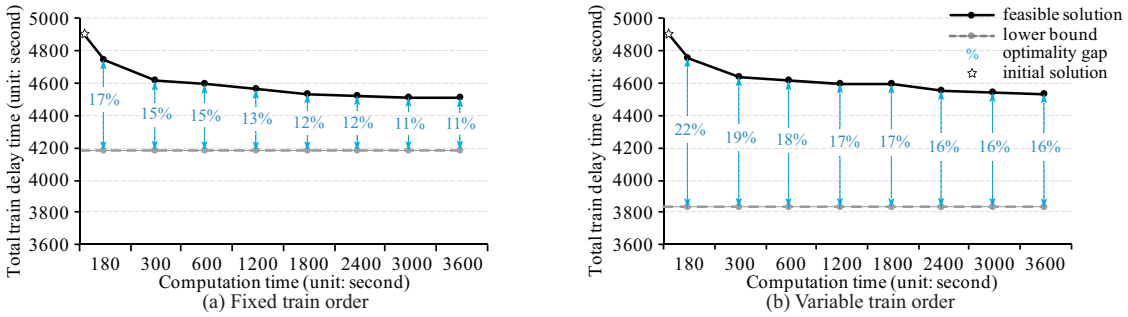


Fig. 12. Lower bounds, feasible solutions, and estimation of optimality gaps

The lower bound of the case with fixed train order in Fig. 12(a) is tighter than that of the case considering variable train order in Fig. 12(b), which results from the reduced solution space by fixing train orders. As shown in Fig. 12(a), when fixing the train orders, the optimality gap is 17% within 180 seconds of computation time, and it is then reduced to 11% by extending the computation time to at most 3600 seconds. In comparison, the optimality gap of the case considering variable train orders is larger, ranging from 22% to 16%, as shown in Fig. 12(b).

5.5. Summary of the experimental results

We here derive the main conclusions, sketched quantitatively in Fig. 13, from the viewpoints of solution feasibility (constraint violation), solution quality, computational efficiency (reported approximately), and applicability for large-scale instances (measured by the total number of the cases, for which at lease one feasible solution is obtained within the given computation time limit). The center indicates the worst performance for all the four items.

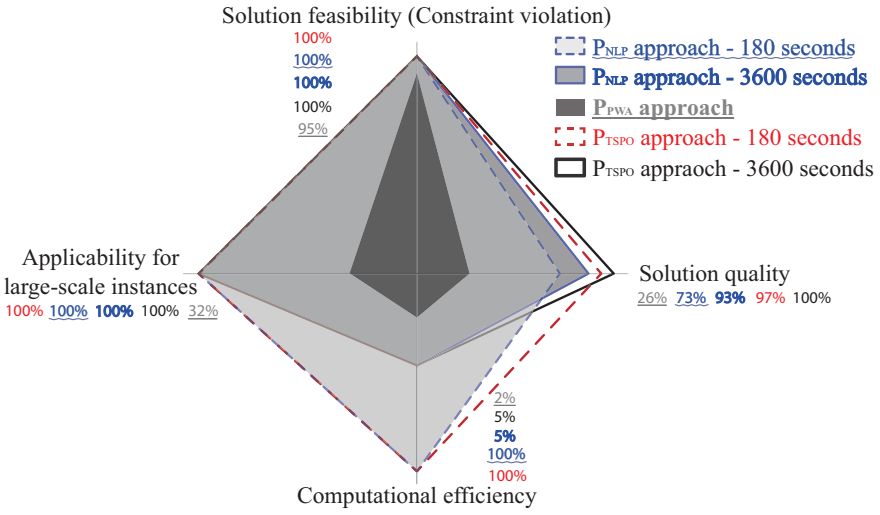


Fig. 13. Overview of the performance of the three proposed approaches

In view of the solution feasibility and the applicability for large-scale instances, the P_{NLP} problem and the P_{TSPO} problem have a similar performance, as they can find feasible solutions for all instances (and

for all delay cases, even the instance with 15 trains). These two approaches perform better than the P_{PWA} problem, because some constraints are violated in the P_{PWA} solution, and for some large-scale instances no feasible solution is obtained by the P_{PWA} problem within the given computation time limit.

Regarding the solution quality, the P_{PWA} approach is also the worst among the three approaches. The solution quality of the P_{NLP} problem and the P_{TSPO} problem differs among instances. The P_{TSPO} approach has a better performance on the instances with a larger number of trains, and the P_{NLP} approach performs a little better on the instances with a smaller number of trains. Overall, the P_{TSPO} solution is better than the P_{NLP} solution, achieving a 23.2% improvement, corresponding to a total delay time of 3727 seconds, within 180 seconds of computation time. The improvement of the P_{TSPO} approach in solution quality reduces to 6.7%, when extending the computation time to 3600 seconds.

From the perspective of computational efficiency, the P_{PWA} approach does not yield any feasible solution within the given time limit for many instances, so the computational efficiency of the P_{PWA} approach is recognized as being the worst. In the experiments, feasible solutions (having satisfactory quality in fact) can always be found by the P_{TSPO} approach within the shortest computation time limit (i.e., 180 seconds), and a significant improvement (with respect to the corresponding initial solution) in solution quality can be achieved efficiently. Regarding the computational efficiency of the P_{NLP} approach, feasible solutions can also be obtained within the given computation time limit, but with a worse quality in comparison with the P_{TSPO} solution. As computation time limits are considered and feasible solutions can be found by both the P_{NLP} approach and the P_{TSPO} approach for all delay cases, within 180 seconds of computation time, we cannot make conclusion on their computational efficiency. Their computational efficiency is therefore reflected by the quality of the solutions obtained within the given computation time limits.

Computational efficiency is a key factor for addressing real-time problems, and the problem of integrating real-time traffic management and train control is such a case. Therefore, the overall performance of the P_{TSPO} approach is recognized as being the best, as a solution with better and satisfactory quality can be found efficiently (within 180 seconds), see Fig. 7. Using a larger set of TSPOs for the P_{TSPO} approach leads to a better solution. The results show that we could consider to fix the train orders when using a larger set of TSPOs, in order to better explore a smaller solution space regarding the train speed within a time limit.

The experimental results demonstrate the benefits of integrating traffic management and train control. The benefit is reflected by the reduced train delays, i.e., train delays can be reduced by managing the train speed and by changing the train orders. In our test case, the consideration of multiple TSPOs leads to 3.14%/8.08% reduction of train delays for Set_1 within 180/3600 seconds of computation time, and the consideration of changing train orders results in an additional 1.59% improvement in the solution quality for Set_2, as discussed in Section 5.4.

6. Conclusions and future research

In this paper, we have tackled the integration of real-time traffic management and train control by using mixed-integer nonlinear programming (MINLP) and mixed-integer linear programming (MILP) methods. Three optimization approaches are developed, i.e., one MINLP problem (P_{NLP}) and two MILP problems (P_{PWA} and P_{TSPO}), for delivering both a train dispatching solution (i.e., binary/integer combinational decisions on a set of times, orders, and routes to be followed by trains) and a train control solution (i.e., train speed trajectories following nonlinear dynamics) simultaneously. In these optimization problems, the train speed is considered variable, and the blocking time of a train on a block section dynamically depends on its real speed. Regarding the solution approaches, we have presented a two-level approach for solving the P_{NLP} problem and proposed a custom-designed two-step approach for solving the P_{TSPO} problem. The performance of the three proposed optimization approaches is comparatively evaluated from the viewpoints of solution feasibility, solution quality, computational efficiency, and applicability for large-scale instances, based on a real-world dataset adapted from the Dutch railway network. According to the experimental results, the P_{TSPO} problem overall yields the best performance among the three optimization approaches, as it is able to exploit the solution space efficiently. Moreover, the benefits of integrating real-time traffic management and train control are demonstrated: the train delay can be reduced up to 8% by managing the train speed and by changing the train orders.

Our future research will focus on the following main extensions. First, although the P_{TSPO} approach is applicable for instances much larger than those in the literature, we could still consider to increase the instance scale. One extension is therefore to apply distributed optimization methods to further improve the computational efficiency, in order to further increase the applicability of the proposed optimization approaches to larger-scale instances. Second, the resistances caused by air, roll, track grade, curves, and tunnels to train movements should be considered to a more precise extent and calibrated as it is prerequisite to accurately estimate the train speed, distance, and headway in areas with a rugged topography. Then, a precise computation of the energy consumption can be derived through considering a precise inclusion of the resistances into cruising. In Part 2 of this paper, we discuss this extension on energy-related aspects, i.e., evaluating the energy consumption for accelerating trains and overcoming resistance and calculating the regenerative energy utilization. Finally, a comprehensive system could be developed based on the P_{TSPO} problem to integrate the multiple steps in the solving procedure, e.g., the preprocessing step for generating a set (or an efficient subset) of the possible TSPOs, the solving step to solve the MILP problem (P_{TSPO}) by using an MILP solver, and the displaying step to show train timetables and speed-space graphs.

Acknowledgments

This work is jointly supported by the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University (RCS2016ZJ003, RCS2016ZT022), National Natural Science Foundation of China (71571012, 61503020). The work of the first author is also supported by China Scholarship Council under Grant 201507090058.

References

Albrecht, A., Howlett, P., Pudney, P., Vu, X., Zhou, P., 2013. Using timing windows to allow energy-efficient driving, in: Proceedings of the 10th World Congress on Rail Research. Sydney, Australia.

Albrecht, T., 2009. The influence of anticipating train driving on the dispatching process in railway conflict situations. Networks and Spatial Economics 9, 85–101.

Albrecht, T., Binder, A., Gassel, C., 2011. An overview on real-time speed control in rail-bound public transportation systems, in: Proceedings of the 2nd International Conference on Models and Technologies for Intelligent Transportation Systems. Leuven, Belgium, pp. 1–4.

Banedanmark, 2017. Report on train punctuality. http://rigsrevisionen.dk/publications/2017/32017/.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., Wagenaar, J., 2014. An overview of recovery models and algorithms for real-time railway rescheduling. Transportation Research Part B: Methodological 63, 15–37.

Caimi, G., Fuchsberger, M., Laumanns, M., Lüthi, M., 2012. A model predictive control approach for discrete-time rescheduling in complex central railway station areas. Computers & Operations Research 39, 2578–2593.

Chevrier, R., Pellegrini, P., Rodriguez, J., 2013. Energy saving in railway timetabling: A bi-objective evolutionary approach for computing alternative running times. Transportation Research Part C 37, 20–41.

Corman, F., D'Ariano, A., Hansen, I.A., Pacciarelli, D., 2011a. Optimal multi-class rescheduling of railway traffic. Journal of Rail Transport Planning & Management 1, 14–24.

Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2009. Evaluation of green wave policy in real-time railway traffic management. Transportation Research Part C: Emerging Technologies 17, 607–616.

Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2010. A tabu search algorithm for rerouting trains during rail operations. Transportation Research Part B: Methodological 44, 175–192.

Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2012. Bi-objective conflict detection and resolution in railway traffic management. Transportation Research Part C: Emerging Technologies 20, 79–94.

Corman, F., D'Ariano, A., Pranzo, M., Hansen, I.A., 2011b. Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. Transportation Planning and Technology 34, 341–362.

Corman, F., Meng, L., 2015. A review of online dynamic models and algorithms for railway traffic management. IEEE Transactions on Intelligent Transportation Systems 16, 1274–1284.

Corman, F., Quaglietta, E., 2015. Closing the loop in real-time railway control: framework design and impacts on operations. Transportation Research Part C: Emerging Technologies 54, 15–39.

D'Ariano, A., Albrecht, T., 2010. Running time re-optimization during real-time timetable perturbations. Part C of Timetable Planning and Information Quality WIT Press, 147–156.

D'Ariano, A., Corman, F., Pacciarelli, D., Pranzo, M., 2008. Reordering and local rerouting strategies to manage train traffic in real time. Transportation Science 42, 405–419.

D'Ariano, A., Pacciarelli, D., Pranzo, M., 2007a. A branch and bound algorithm for scheduling trains in a railway network. European Journal of Operational Research 183, 643–657.

D'Ariano, A., Pranzo, M., Hansen, I.A., 2007b. Conflict resolution and train speed coordination for solving real-time timetable perturbations. IEEE Transactions on Intelligent Transportation Systems 8, 208–222.

Dollevoet, T., Huisman, D., Schmidt, M., Schöbel, A., 2012. Delay management with rerouting of passengers. Transportation Science 46, 74–89.

Fang, W., Yang, S., Yao, X., 2015. A survey on problem models and solution approaches to rescheduling in railway networks. IEEE Transactions on Intelligent Transportation Systems 16, 2997–3016.

Ginkel, A., Schöbel, A., 2007. To wait or not to wait? The bicriteria delay management problem in public transportation. Transportation Science 41, 527–538.

Hansen, I.A., Pachl, J., 2014. Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation. Eurailpress, Hamburg, Germany.

Howlett, P., 2000. The optimal control of a train. Annals of Operations Research 98, 65–87.

Howlett, P.G., Pudney, P.J., 2012. Energy-Efficient Train Control. Springer Science & Business Media.

INFORMS RAS, 2012. Institute for Operations Research and Management Sciences (INFORMS) Railroad Application Section (RAS) problem solving competition. http://connect.informs.org/railway-applications/awards/problem-solving-competition/2012.

Kecman, P., Corman, F., D'Ariano, A., Goverde, R.M., 2013. Rescheduling models for railway traffic management in large-scale networks. Public Transport 5, 95–123.

Li, X., Lo, H.K., 2014a. An energy-efficient scheduling and speed control approach for metro rail operations. Transportation Research Part B: Methodological 64, 73–89.

Li, X., Lo, H.K., 2014b. Energy minimization in dynamic train scheduling and control for metro rail operations. Transportation Research Part B: Methodological 70, 269–284.

Lüthi, M., 2009. Improving the efficiency of heavily used railway networks through integrated real-time rescheduling. Ph.D. thesis. ETH Zürich.

Mazzarello, M., Ottaviani, E., 2007. A traffic management system for real-time traffic optimisation in railways. Transportation Research Part B: Methodological 41, 246–274.

Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an n-track network: A model reformulation with network-based cumulative flow variables. Transportation Research Part B: Methodological 67, 208–234.

Montigel, M., 2009. Operations control system in the Lötschberg base tunnel. Railway Technical Review 2, 43–44.

Mu, S., Dessouky, M., 2011. Scheduling freight trains traveling on complex networks. Transportation Research Part B: Methodological 45, 1103–1123.

Narayanaswami, S., Rangaraj, N., 2011. Scheduling and rescheduling of railway operations: a review and expository analysis. Technology Operation Management 2, 102–122.

Network Rail, 2017. Punctuality on the national rail network. https://www.networkrail.co.uk/who-we-are/how-we-work/performance/public-performance-measure/punctuality-national-rail-network/.

On-Time, 2014. Optimal networks for train integration management across Europe. http://www.ontime-project.eu/aboutproject.aspx.

Pachl, J., 2009. Railway Operation and Control (2nd edn). VTD Rail Publishing, Mountlake Terrace.

Pellegrini, P., Marlière, G., Pesenti, R., Rodriguez, J., 2015. RECIFE-MILP: An effective MILP-based heuristic for the real-time railway traffic management problem. IEEE Transactions on Intelligent Transportation Systems 16, 2609–2619.

Pellegrini, P., Marlière, G., Rodriguez, J., 2014. Optimal train routing and scheduling for managing traffic perturbations in complex junctions. Transportation Research Part B: Methodological 59, 58–80.

Quaglietta, E., Corman, F., Goverde, R.M., 2013. Stability analysis of railway dispatching plans in a stochastic and dynamic environment. Journal of Rail Transport Planning & Management 3, 137–149.

Quaglietta, E., Pellegrini, P., Goverde, R.M., Albrecht, T., Jaekel, B., Marlière, G., Rodriguez, J., Dollevoet, T., Ambrogio, B., Carcasole, D., et al., 2016. The on-time real-time railway traffic management framework: A proof-of-concept using a scalable standardised data communication architecture. Transportation Research Part C: Emerging Technologies 63, 23–50.

Rao, X., Montigel, M., Weidmann, U., 2013. Potential railway benefits according to enhanced cooperation between traffic management and automatic train operation, in: Proceedings of the 2013 IEEE International Conference on Intelligent Rail Transportation (ICIRT). Beijing, China, pp. 111–116.

Research Collection ETH Zurich, 2018. Detailed experimental results. https://www.research-collection.ethz.ch/handle/20.500.11850/256447.

Rodrigo, E., Tapia, S., Mera, J., Soler, M., 2013. Optimizing electric rail energy consumption using the lagrange multiplier technique. Journal of Transportation Engineering 139, 321–329.

Rodriguez, J., 2007. A constraint programming model for real-time train scheduling at junctions. Transportation Research Part B: Methodological 41, 231–245.

Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., Pacciarelli, D., 2016. Ant colony optimization for the real-time train routing selection problem. Transportation Research Part B: Methodological 85, 89–108.

Schachtebeck, M., Schöbel, A., 2010. To wait or not to wait-and who goes first? Delay management with priority decisions. Transportation Science 44, 307–321.

Schöbel, A., 2007. Integer Programming Approaches for Solving the Delay Management Problem. Springer.

Törnquist, J., Persson, J.A., 2007. N-tracked railway traffic re-scheduling during disturbances. Transportation Research Part B: Methodological 41, 342–362.

Törnquist, K.J., 2012. Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. Transportation research. Part C, Emerging technologies 20, 62–78.

Tuyttens, D., Fei, H., Mezmaz, M., Jalwan, J., 2013. Simulation-based genetic algorithm towards an energy-efficient railway traffic control. Mathematical Problems in Engineering 2013.

V/Line, 2017. Performance and capacity. https://www.vline.com.au/About-V-Line/Performance.

Wang, P., Goverde, R.M., 2016. Multiple-phase train trajectory optimization with signalling and operational constraints. Transportation Research Part C: Emerging Technologies 69, 255–275.

Wang, Y., De Schutter, B., van den Boom, T.J.J., Ning, B., 2013. Optimal trajectory planning for trains–a pseudospectral method and a mixed integer linear programming approach. Transportation Research Part C: Emerging Technologies 29, 97–114.

Wang, Y., Ning, B., van den Boom, T.J.J., De Schutter, B., 2016. Optimal Trajectory Planning and Train Scheduling for Urban Rail Transit Systems. Springer.

Wang, Y., Ning, B., Cao, F., De Schutter, B., van den Boom, T.J.J., 2011. A survey on optimal trajectory planning for train operations, in: Proceedings of the 2011 IEEE International Conference on Service Operations, Logistics, and Informatics (SOLI). Beijing, China, pp. 589–594.

Williams, H.P., 2013. Model building in mathematical programming. John Wiley & Sons.

Xu, P., Corman, F., Peng, Q., Luan, X., 2017. A train rescheduling model integrating speed management during disruptions of high-speed traffic under a quasi-moving block system. Transportation Research Part B: Methodological 104, 638 – 666.

Yang, X., Li, X., Ning, B., Tang, T., 2016. A survey on energy-efficient train operation for urban rail transit. IEEE Transactions on Intelligent Transportation Systems 17, 2–13.

Yang, X., Ning, B., Li, X., Tang, T., 2014. A two-objective timetable optimization model in subway systems. IEEE Transactions on Intelligent Transportation Systems 15, 1913–1921.

Zhan, S., Kroon, L.G., Veelenturf, L.P., Wagenaar, J.C., 2015. Real-time high-speed train rescheduling in case of a complete blockage. Transportation Research Part B: Methodological 78, 182–201.

Zhou, L., Tong, L., Chen, J., Tang, J., Zhou, X., 2017. Joint optimization of high-speed train timetables and speed profiles: A unified modeling approach using space-time-speed grid networks. Transportation Research Part B: Methodological 97, 157 – 181.

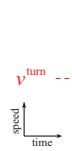
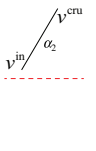
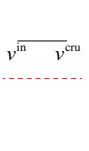
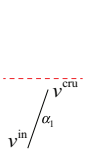
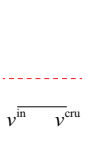
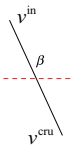
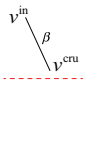
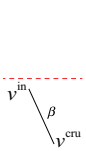
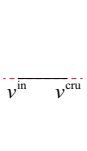
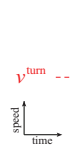
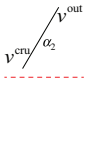
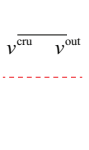
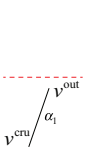
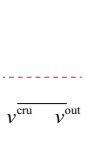
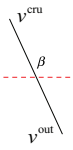
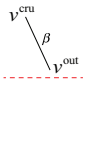
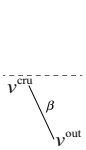
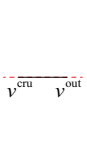
Appendix A Additional explanations of the formulations in Section 3

In Section 3, we have introduced six logical speed indicators $\zeta_{1,f,i,j}$, ..., $\zeta_{6,f,i,j}$ to indicate the actions taken by train f on cell (i,j), i.e., the train trajectory. Some constraints, e.g., (25) and (27), further employ these indicators to perform their functions. For assisting the readers to understand our formulations, we here describe the six logical speed indicators in detail, and then we explain how these indicators play a role in other constraints. In the remainder of this section, we omit the subscripts f,i,j of the parameters and variables to improve the readability, e.g., the incoming speed is denoted as v^{in}, and the acceleration is indicated as α_1 when the train speed is less than the switching speed v^{turn} (the speed point for switching the train acceleration) and as α_2 when the train speed is larger than the switching speed v^{turn}.

A.1 Explanation of the six logical speed indicators $\zeta_{1,f,i,j}, ..., \zeta_{6,f,i,j}$ in Table 3

Table 6 summarizes all possible train trajectories, i.e., the action(s) that a train may take, in the incoming and outgoing phases respectively.

Table 6. Summary of the possible train trajectories and the corresponding value of the speed indicators

	Trajectory ID	Trajectory_1	Trajectory_2	Trajectory_3	Trajectory_4	Trajectory_5	Trajectory_6	Trajectory_7	Trajectory_8	Trajectory_9
Incoming phase										
	ζ_1	1	1		1		0	0	0	1
	ζ_3	0	1		0		1	1	0	1
	ζ_4	0	0		1		1	0	1	1
	Trajectory ID	Trajectory_10	Trajectory_11	Trajectory_12	Trajectory_13	Trajectory_14	Trajectory_15	Trajectory_16	Trajectory_17	Trajectory_18
Outgoing phase										
	ζ_2	1	1		1		0	0	0	1
	ζ_5	0	1		0		1	1	0	1
	ζ_6	0	0		1		1	0	1	1

As presented, there are 9 possible trajectories for each phase. Each scenario can be represented by the speed indicators ζ_1, ζ_3, and ζ_4 for the incoming phase or by the speed indicators ζ_2, ζ_5, and ζ_6 for the outgoing phase. Regarding the cruising phase, the train speed is constant, so only one train trajectory is possible, like "Trajectory_3", "Trajectory_5", and "Trajectory_9".

A.2 Explanation of (25)

Constraints (25a)-(25e) are proposed for the incoming phase by employing the speed indicators and by satisfying the formula of the uniformly accelerating and decelerating motions, i.e., for such a motion with an initial speed v_0, a final speed v_t, and a steady acceleration α, the elapsed time for accelerating from speed v_0 to speed v_t is $\Delta t = \frac{v_t - v_0}{\alpha}$. As shown in Table 7, constraints (25a)-(25e) represent the 9 possible trajectories for the incoming phase in Table 6.

Regarding the cases of "Trajectory_3", "Trajectory_5", and "Trajectory_9", as the incoming speed v^{in} equals the cruising speed v^{cru}, the incoming phase does not exist anymore, and the condition $a^{\text{cru}} = a$ is required by (25b) and (25c). Note that similar constraints can be constructed to represent the "Trajectory_10", ..., "Trajectory_18" for the outgoing phase in Table 6. We do not present those details here.

Table 7. Overview of the details of (25)

Constraint	Corresponding trajectory ID	Value of the speed indicators ζ_1	ζ_3	ζ_4	Reduced equation
(25a)	Trajectory_6, Trajectory_7, and Trajectory_8	0	0 or 1	0 or 1	$a^{\text{cru}} - a = -\frac{v^{\text{cru}} - v^{\text{in}}}{\beta}$
(25b)	Trajectory_2, Trajectory_3, and Trajectory_9	1	1	0 or 1	$a^{\text{cru}} - a = \frac{v^{\text{cru}} - v^{\text{in}}}{\alpha_2}$
(25c)	Trajectory_4, Trajectory_5, and Trajectory_9	1	0 or 1	1	$a^{\text{cru}} - a = \frac{v^{\text{cru}} - v^{\text{in}}}{\alpha_1}$
(25d)	Trajectory_1	1	0	0	$a^{\text{turn}} - a = \frac{v^{\text{turn}} - v^{\text{in}}}{\alpha_1}$
(25e)	Trajectory_1	1	0	0	$a^{\text{cru}} - a^{\text{turn}} = \frac{v^{\text{cru}} - v^{\text{turn}}}{\alpha_2}$

A.3 Explanation of (27)

Constraints (27a)-(27d) are proposed for calculating the distance L^{in} that a train travels within a cell in the incoming phase. These constraints also satisfy the formula of the uniformly accelerating and decelerating motions, i.e., for such a motion with an initial speed v_0, a final speed v_t, and a steady acceleration α, the distance traveled for accelerating from speed v_0 to speed v_t is $L = \frac{{v_t}^2 - {v_0}^2}{2 \cdot \alpha}$. As shown in Table 8, constraints (27a)-(27d) represent the 9 possible trajectories for the incoming phase in Table 6.

Table 8. Overview of the details of (27)

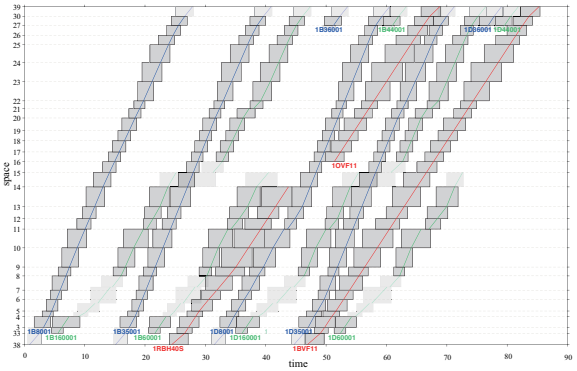
Constraint	Corresponding trajectory ID	Value of the speed indicators ζ_1	ζ_3	ζ_4	Reduced equation
(27a)	Trajectory_6, Trajectory_7, and Trajectory_8	0	0 or 1	0 or 1	$L^{\text{in}} = -\frac{(v^{\text{cru}})^2 - (v^{\text{in}})^2}{2 \cdot \beta}$
(27b)	Trajectory_2, Trajectory_3, and Trajectory_9	1	1	0 or 1	$L^{\text{in}} = \frac{(v^{\text{cru}})^2 - (v^{\text{in}})^2}{2 \cdot \alpha_2}$
(27c)	Trajectory_4, Trajectory_5, and Trajectory_9	1	0 or 1	1	$L^{\text{in}} = \frac{(v^{\text{cru}})^2 - (v^{\text{in}})^2}{2 \cdot \alpha_1}$
(27d)	Trajectory_1	1	0	0	$L^{\text{in}} = \frac{(v^{\text{turn}})^2 - (v^{\text{in}})^2}{2 \cdot \alpha_1} + \frac{(v^{\text{cru}})^2 - (v^{\text{turn}})^2}{2 \cdot \alpha_2}$

Regarding the "Trajectory_3", "Trajectory_5", and "Trajectory_9", as the incoming speed v^{in} equals the cruising speed v^{cru}, the incoming phase does not exist anymore, and then the distance L^{in} equals zero according to (27b) and (27c). Note that similar constraints can be constructed to represent the "Trajectory_10", ..., "Trajectory_18" in Table 6, for calculating the distance L^{out} that a train runs over on a cell in the outgoing phase. We do not present those details here.

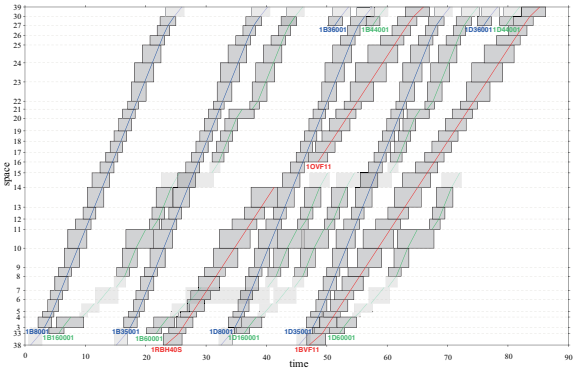
Appendix B Illustration of the train timetables

We report here the train timetables of a representative case for the Dutch test case (regarding the experiments in Section 5.2), obtained by the P_{NLP} problem (in Fig. 14(a)) and the P_{TSPO} problem (an initial solution in Fig. 14(b) and a secondary solution in Fig. 14(c)) respectively. Fig. 14(d) then provides the speed-space graphs for all trains, corresponding to the train timetables given in Fig. 14(a)-Fig. 14(c). As there are siding tracks in some station areas, it is hard to draw every train path in a single timetable. In order to present all train paths completely, we draw the train blocking times on the main tracks by using dark gray blocks, and we use light gray blocks to show the train blocking times on the siding tracks. Therefore, an overlap of the dark and light gray blocks does not indicate a train conflict, and it means that the two trains are running on different siding tracks in the same station area.

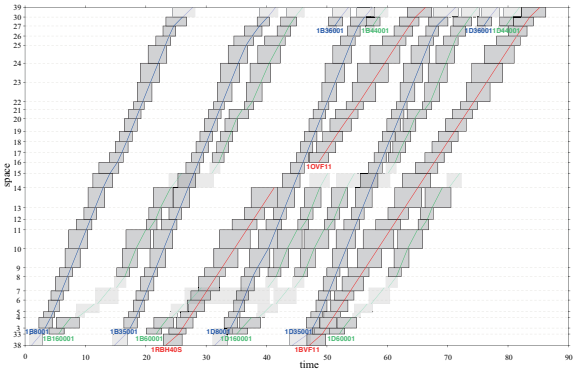
The total train delay time of the train timetables in Fig. 14(a)-Fig. 14(c) is 3993 seconds, 3793 seconds, and 3426 seconds respectively. As we can see in the train timetables of Fig. 14(a) and Fig. 14(b)-Fig. 14(c), the orders of the sprinter train 1B60001 and the intercity train 1D8001 (and the freight train 1RBH40S as well) change on some cells, e.g., cell (8, 9). As a result, in Fig. 14(b), the sprinter train 1B60001 has more delays (916 seconds), and the sum of the delays of the other affected trains (including train 1RBH40S, 1D8001, and 1OVF11) decreases by 1219 seconds; in Fig. 14(c), the delay of train 1B60001 increases by 927 seconds, and the total delay of the other affected trains decreases by 1302 seconds.



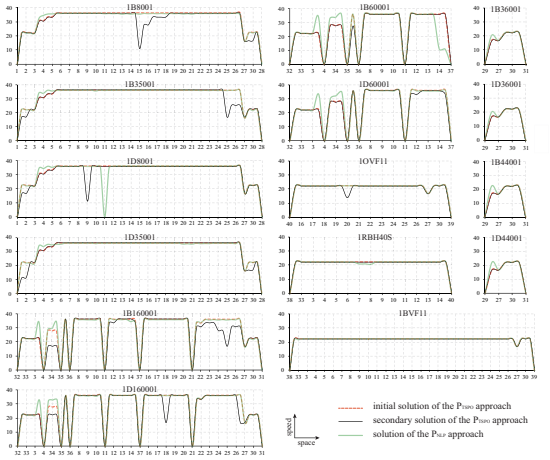
(a) Train timetable, corresponding to the solution obtained by the P_{NLP} problem



(b) Train timetable, corresponding to the initial solution of the P_{TSPO} problem



(c) Train timetable, corresponding to the secondary solution of the P_{TSPO} problem



(d) Speed-space graphs for all trains, corresponding to the train timetables in (a)-(c)

Fig. 14. Train timetables and train speed-space graphs for the Dutch railway network

Appendix C Case study based on the railway network from the INFORMS RAS problem solving competition 2012

C.1 Description of the railway network

To further assess the model performance on larger-scale instances, we adapt the railway network from the INFORMS RAS problem solving competition 2012 (INFORMS RAS 2012), with both single-track segments and double-track segments, consisting of 67 nodes and 76 cells, as sketched in Fig. 15(a).

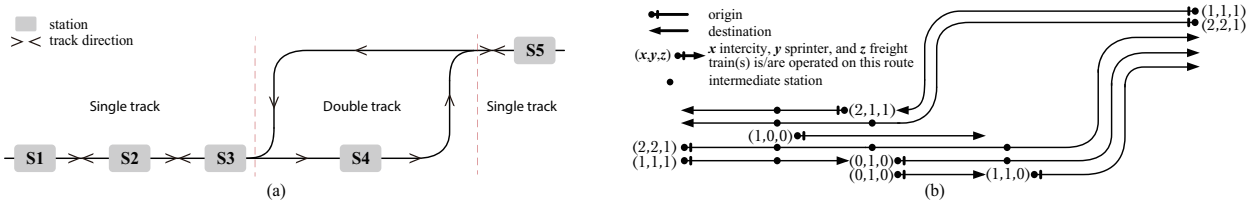


Fig. 15. A rail network adapted from INFORMS RAS (2012)

The train data (e.g., acceleration/deceleration rate, category, and length) and the stop pattern same to the Dutch railway network are used here; we refer to Section 5.1 for more information. We consider 2.5 hours of traffic with 25 trains, including 10 intercity, 10 sprinter, and 5 freight trains, and six global (bi-)directional train routes, as illustrated in Fig. 15(b). Each route has a mark in the form of (x, y, z) at its origin; the mark indicates the numbers of intercity (x), sprinter (y), and freight (z) trains respectively that are operated on this route.

C.2 Performance of the $\mathrm{P}_{\mathrm{TSPO}}$ *model on a larger-scale instance*

As evaluated in Section 5.2, the $\mathrm{P}_{\mathrm{TSPO}}$ model yields the best performance, and the other two models already have computation burden in the experiments based on the Dutch railway network, either obtaining no feasible solution or taking a much longer computation time. In this section, we only examine the $\mathrm{P}_{\mathrm{TSPO}}$ model performance on larger-scale instances, by using the INFORMS RAS railway network described in Section C.1. We use the larger set of TSPOs (i.e., Set_1 in Table 5), due to its good solution quality, as discussed in Section 5.4. The average results of the 10 delay cases with randomly generated primary delays are illustrated in Fig. 16, including the initial solution, the secondary solutions along the computation time, and the improvement in the objective value.

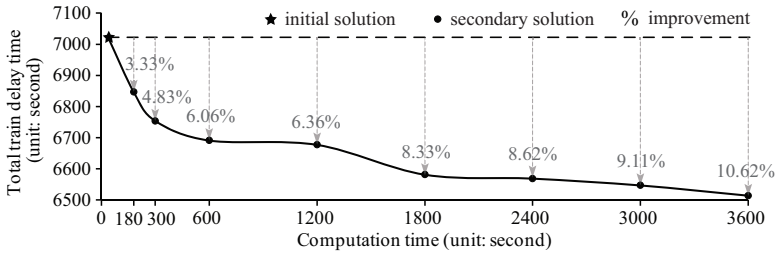
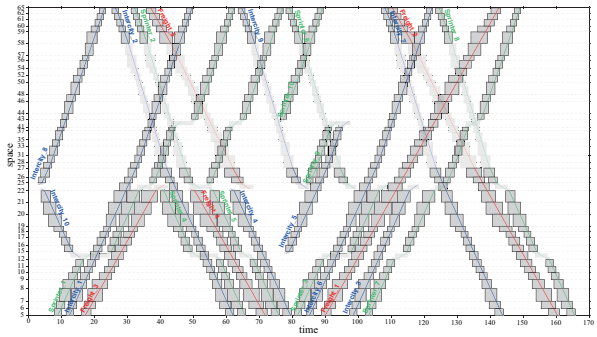
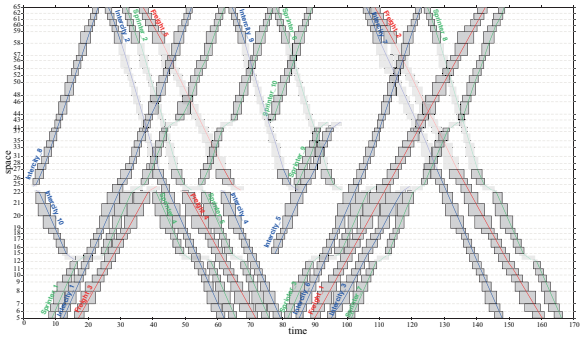


Fig. 16. Total train delay time as a function of computation time, results for the INFORMS RAS railway network

Similar to the results of the Dutch railway network, the initial solution is still obtained very quickly, and the total train delay time decreases as a function of the computation time in the secondary solutions. Considering multiple TSPOs achieves 3.33% improvement in the train delay time within 180 seconds, which increases to 10.62% when the computation time is extended to 3600 seconds.



(a) Train timetable, corresponding to the initial solution of the $\mathrm{P}_{\mathrm{TSPO}}$ model on INFORMS RAS railway network



(b) Train timetable, corresponding to the secondary solution of the $\mathrm{P}_{\mathrm{TSPO}}$ model on INFORMS RAS railway network

Fig. 17. Train timetables for the INFORMS RAS railway network

Fig. 17 reports the train timetables of a representative case for the INFORMS RAS railway network, obtained by the $\mathrm{P}_{\mathrm{TSPO}}$ model. An initial solution and a secondary solution are provided in Fig. 17(a) and Fig. 17(b) respectively.