Optimization model and algorithm for integrating train timetabling and track maintenance task scheduling

Yongxiang Zhang[1], Andrea D'Ariano[2], Bisheng He [1,*], Qiyuan Peng [1]

1 Southwest Jiaotong University, School of Transportation and Logistics, P.O. Box 610031, Chengdu, China

2 Roma Tre University, Department of Engineering, Via della Vasca Navale 79, 00146, Roma, Italy

* Corresponding author email: bishenghe@swjtu.edu.cn

Abstract

This paper addresses the problem of improving the integration between passenger timetabling and track maintenance scheduling. We propose a microscopic optimization model and an iterative algorithm for solving this problem efficiently. Block sections are considered as the basic microscopic elements for train movements in a railway network. A mixed-integer linear programming formulation is proposed for the integrated optimization problem in which train timing, sequencing and routing are the timetabling variables, while timing and sequencing of maintenance tasks are the track maintenance variables. The constraints and objective function proposed in this work address the practical specifications of the INFORMS RAS 2016 Problem Solving Competition (2016 PSC). In this context, the main decision variables are the entrance and exit times of the trains on each block section plus the start and end times of each maintenance task. Since the integrated optimization problem is strongly NP-hard, an iterative algorithm is proposed to compute near-optimal solutions in a short computation time. The algorithm is based on a decomposition of the overall problem into sub-problems related to train scheduling and/or routing with or without track maintenance task scheduling. The connecting information between the two sub-problems concerns the selected train routes plus the start and end times of the maintenance tasks. Computational experiments are performed on a set of realistic railway instances, which were introduced during the 2016 PSC. The iterative algorithm outperforms a standard MILP solver and the first-place team of this competition in terms of both solution quality and time to deliver the new best-known solutions.

Keywords: Timetabling; Infrastructure Maintenance; Integrated Optimization; Mixed-Integer Linear Programming.

1 Introduction

The Track Maintenance Task Scheduling (TMTS), also known as the railway infrastructure maintenance planning, is the problem of finding a feasible schedule of the maintenance activities, or tasks, to retain the railway track in appropriate state. This is required to provide sufficient availability of railway capacity for allocating the required railway services. Infrastructure maintenance consumes a large amount of the railway budget to manage vehicles, teams, and materials used for it (CER 2012). Furthermore, maintenance activities reduce the available railway capacity since no train can occupy the maintenance tracks during the work time. Most railway companies thus adopt a detailed process for planning the maintenance schedules to achieve efficient and effective plans (Budai et al. 2006, Lidén 2015).

The planning of track maintenance activities is interrelated with the Train Timetabling (TT) problem which schedules the movement of trains on the tracks. Optimizing the TT problem is considered as a cost-effective way to expand the railway capacity and earn revenue. However, the planning conflicts between the TT and TMTS problems are obviously critical, especially in high-density railway networks (D'Ariano et al. 2017, Lidén et al. 2018).

With the increasing demand on passenger and freight transportation in the recent years (S2R 2015), more and more trains need to be inserted in the timetable, and the tracks would need to be used more frequently, leading to a higher probability of track failures. That trend thus requires additional maintenance activities on the tracks, which may be unavailable for trains for larger time periods, thus causing alternations of the timetable, such as train delays or even service cancellations. Hence, TT and TMTS problems must be managed simultaneously to solve potential planning conflicts and to allocate the available railway capacity more efficiently to the train services in the timetable.

The recent development of computer-aided railway systems, such as centralized traffic control systems (Cui and Zhuang 2013), enables the possibility to improve the integration between the TT and TMTS problems. However, in practice, these two problems are designed by different competing departments. Furthermore, the literature about train timetabling focuses on determining train timing, sequencing and routing on the tracks (Cordeau et al. 1998, Cacchiani et al. 2014A-2014B), while the maintenance tasks are designed before the train timetabling process starts. These tasks are thus set as fixed information to the train timetabling problem. A comprehensive survey on the railway infrastructure maintenance (Lidén, 2015) also shows that train timetabling and track maintenance task scheduling are often investigated separately, although these are strongly related in practice.

Recently, the integration of TT and TMTS received the attention of some researchers (D'Ariano et al. 2017, Luan et al. 2017, Lidén et al. 2018, Meng et al. 2018). Furthermore, since this integration clearly deserves further research, INFORMS Railway Application Section (RAS) proposed a Problem Solving Competition in 2016 (2016 PSC) on the following topic: *"Routing Trains through a Railway Network: Joint optimization on train timetabling and maintenance task scheduling"*. The 2016 PSC aims to solve the train routing problem through a complex railway network in combination with infrastructure maintenance decisions. In total, 45 teams from around the world participated in the competition from 11 countries around the world. Our paper focuses on improving the best results obtained during the 2016 PSC, i.e. on developing models and algorithms for the optimal integration of TT and TMTS at a tactical level.

The current challenging issues for the development of an integrated method for TT and TMTS are the following:

(1) The train movement and maintenance task on each track must be designed at a microscopic level in order to carefully model all the relevant aspects of the integrated problem. As the TT and TMTS decisions must be taken simultaneously, the train routing decisions consider the status of microscopic infrastructure elements. However, as noticed in Cacchiani et al. (2014A-2014B), the train timetabling problem is most often modelled at a macroscopic level. Only few of researchers focus on the development of microscopic TT methods. The main reason is related to the size of the mathematical model that increases dramatically, in terms of decision variables and constraints, as soon as the level of modelling detail increases. In addition, accurate timing of the activities (at the level of detail of seconds) needs to be provided to improve the allocation of railway capacity. This accuracy cannot be easily provided by time-decomposition approaches, since the time horizon of optimization needs to be discretized into several small-size time intervals.

(2) All the infrastructure maintenance constraints must be already satisfied at a tactical level, since the violation of maintenance requirements can only be recovered in a later stage at a high operational cost, e.g. with the cancellation of train services. Some practical maintenance constraints involve strong safety requirements for the train movements, such as speed limitations in some tracks during and after maintenance works. Furthermore, this type of constraints is very complex and can be hardly formulated in a linear way. In this paper, as designed by practitioners for the 2016 PSC, we consider speed restriction constraints for the first two trains running in a track after the completion of each maintenance task and speed reduction constraints for the trains running in the track during the time window in which an opposite-direction cell of this track is being maintained. A special speed trajectory profile is required for these trains in order to check the current status of the track and avoid any potential risk of failures. Even the mostly related papers in the literature, e.g. Luan et al. (2017) and D'Ariano et al. (2017), did not consider these requirements in their models.

(3) The optimal integration of TT and TMTS is a difficult problem, since both TT and TMTS are NP-Hard problems (Fang et al. 2015, Lidén 2015). The problem complexity increases rapidly with the problem size and the instances provided for the 2016 PSC are challenging. This is proved by the long computation time (more than 1 hour) required by all the finalist teams in 2016 PSC (Carpov et al. 2016, Wang et al. 2016, and He et al. 2016) in order to compute a near-optimal solution to each instance provided by the organizers of the 2016 PSC.

This paper contributes to face the three above methodological issues on the integration of TT and TMTS. The original contribution of our work is the development of a mathematical model and an efficient solution method.

As for point 1, we consider the cell as the atomic resource to be managed for detecting potential conflicts, meaning that if a link (a piece of track) is occupied then all other links in the same cell will be occupied as well. Cells are grouped into block sections that are the basic microscopic elements for train movements in a railway network (Hansen and Pachl, 2014). In this paper, we formulate the scheduling decisions at the level of block sections and formulate the integrated optimization problem as a Mixed-Integer Linear Programming (MILP) formulation in which train timing, sequencing and routing are the timetabling variables, while timing and sequencing of maintenance tasks are the track maintenance variables. Specifically, we use the big-*M* method to model train timetabling and maintenance task scheduling decisions.

As for point 2, the proposed big-*M* MILP formulation is customized to deal with the objective function, all the infrastructure maintenance constraints and all the other constraints required by the 2016 PSC, including constraints on the train movement, train route, block section occupancy, maintenance task scheduling, maintenance speed restriction and maintenance speed reduction. The latter constraints require speed profile adjustments for the first and second trains that are scheduled after each maintenance task and for the trains running in the opposite direction with respect to the given maintenance tasks. All the constraints can be modeled in a linear way and at a microscopic level (point 1).

As for point 3, an iterative algorithm is proposed to compute near-optimal solutions in a short computation time. The algorithm is based on a decomposition of the overall problem into two sub-problems: (i) the optimization of train scheduling with track maintenance task scheduling constraints, (ii) the optimization of train scheduling and (re)routing. The connecting information between the sub-problems (i) and (ii) concerns the routing of the trains in the network plus the start and end times of the maintenance tasks. The proposed algorithm is compared quantitatively with a standard MILP solver and with the results obtained by the first-place team in 2016 PSC.

The next sections of the paper are organized as follows. Section 2 provides a literature review on TT and TMTS problems. Section 3 describes the railway network components, problem definitions and preliminaries. Section 4 presents the mathematics model proposed in this paper for the integration of TT and TMTS problems. Section 5 explains the iterative algorithm to solve the integrated problem. Section 6 shows the computational results on the proposed methods and a detailed comparison with other methods, with the aim to proof the validity of our approach. Section 7 summarizes our contributions to the literature and outlines directions for further research on the integrated problem. Appendix shows the new best-known solutions regarding the 2016 PSC computed by our methodological approach.

2 Literature review

This section provides a brief review on the methods proposed in the literature to model and solve the TT and TMTS problems and to integrate various aspects of these two problems.

The train timetabling problem is a core problem in railway planning process. This is a typical NP-hard problem (Caprara et al. 2002), which requires to compute a conflict-free movement for each train in the network and to specify train routes, sequences and detailed arrival and departure times in each relevant resource. The key issue for the computation of conflict-free schedules is the satisfaction of the safety requirements between consecutive trains. This issue requires to deal with the strong constraint that trains must not share the same railway resources simultaneously. The literature on the TT problem (also considering related problems such as train scheduling, train dispatching, and track allocation problem) is quite large. Detailed reviews on the TT problem can be found, e.g., in Cordeau et al. (1998), Zhou and Zhong (2007), Cacchiani et al. (2014A-2014B), Corman and Meng (2014), Fang et al. (2015).

The routing of trains is modelled in the TT problem at macroscopic or microscopic levels, depending on the level of detail of the network representation in the mathematical model. As described in the survey papers of Cacchiani et al. 2014A-2014B, most of the TT literature approaches this problem at a macroscopic level, since only arrival and departure times are designed in each station and running times of the trains are computed between consecutive stations. However, if the train routes are defined at a microscopic level, the running times and minimum required headway times between consecutive trains can be modelled with a significantly higher degree of accuracy, while the available railway infrastructure can be better assigned to the trains in a highly efficient way in the timetabling process. This improvement is more evident when scheduling trains in highly dense and complex railway areas. The recent trend is thus to model the TT problem as much as possible at a microscopic level. With the aim to compute an optimized timetable for a practical-size network, Bešinović et al. (2016) combine microscopic and macroscopic models with a hierarchical framework, where the microscopic model computes minimum running times and headways for each individual train, while the macroscopic model takes this information as input to compute the overall timetable.

Fang et al. (2015) summarize the constraints and objectives related to typical mathematical models, including constraint programming (Rodriguez, 2007), disjunctive programming (D'Ariano et al. 2007), set-packing inspired methods (Caimi et al. 2011). The most popular approaches are the following: 1) job-shop scheduling formulations, 2) time-space network-based formulations, 3) big-*M* formulations. We next review these three types of approaches.

The job-shop scheduling approach determines the train sequencing and timing on each relevant railway resource. This approach can be modelled via disjunctive graphs (Higgins et al. 1997, Burdett and Kozan, 2010, Liu and Kozan 2010) or alternative graphs (D'Ariano et al. 2007, Corman et al. 2009-2010, Samà et al. 2017a). The trains are considered as the jobs and the railway resources (i.e. station platforms, track segments, block sections, tracks) are considered as the machines. Train scheduling problems can be modelled by this type of approach and solved via exact methods, meta-heuristic algorithms or heuristic rules (as provided in D'Ariano et al. 2007, Liu and Kozan 2010). D'Ariano et al. (2008), Corman et al. (2010), Larsen et al. (2014), Samà et al. (2016-2017a-2017c) propose train scheduling and routing frameworks to improve the quality the train schedules by optimizing the selection of train routes. When dealing with practical-size instances, to overcome the issue of a huge solution space, D'Ariano and Pranzo (2009) and Samà et al. (2014) present rolling time horizon approaches combined with the alternative graph model. However, due to the structure of this approach, the rolling time horizon method delivers a local optimal solution since it can only optimize a subset of the decision variables in each time horizon. Recently, Samà et al. (2017a) introduced a hybridization of variable neighborhood search and tabu search methods to solve a train re-routing problem at a microscopic level and to find near-optimal railway traffic management solutions.

Approaches based on time-space or similar networks are alternative ways to describe the movement of trains and to resolve conflicting decisions. Caprara et al. (2002-2006) introduce a graph theory-based model in the macroscopic level with a Lagrangian relaxation algorithm to handle the station capacity constraints. The same time-space network is considered in Cacchiani et al. (2008). Peng et al. (2010) propose another time-space network model with consideration of preventive maintenance constraints. Lusby et al. (2011) use a time-space network to find optimal train paths in the junctions with trains moving at different speeds in a microscopic network. Schlechte (2012) consider the real-world track allocation problem for the railway system. The time expanded train scheduling digraph is proposed to build a coupling mathematical models by a micro-macro transformation. Meng and Zhou (2014) implement a time-space network to optimize train rerouting and rescheduling simultaneously with a Lagrangian relaxation algorithm. He et al. (2014) solve the train timetabling problem with a branch-and-price algorithm and show that the shortcoming of the time-space network about time discretization. When the problem is divided into small intervals, in the order of seconds, the identification of the optimal solution for this type of modelling approach usually requires a long computation time. This is due to the complexity of practical-size TT problems. Specifically, it is hard for a time-space network method to obtain the continuous departure and arrival times of the trains, in case of both the macroscopic or microscopic levels.

Another approach to handle the conflict-free train scheduling constraints is the big-*M* modelling method. Higgins et al. (1997) and Dessouky et al. (2006) develop big-*M* based formulations for the train scheduling problem with fixed train routes. In these two papers, the problem is solved by branch and bound procedures in which the key variables are the train sequencing variables. Törnquist and Persson (2007) use the big-*M* method to determine the sequence of events and their re-scheduling in presence of disturbances, while the routing of trains is also optimized in an *N*-tracked network. Time discretization strategies are used to develop heuristic measures for the sequencing of trains on the rail segments. Yan and Yang (2011) introduce 0-1 decision variables to model the train sequencing with a pair of big-M constraints on multi-track territories. They also adopt a time decomposition method to solve the problem. Mu and Dessouky (2011) add train routing flexibility into a similar big-M formulation to manage a complex railway network efficiently, use a genetic algorithm to identify promising train routes, and solve the resulting train scheduling problem with fixed routing. Pellegrini et al. (2014-2015) consider the timetable rescheduling problem with consideration of train re-routing variables. A two-step approach is proposed to solve the problem. In the first step, a MILP solver optimizes the train scheduling with fixed routes. In the second step, the solver also optimizes train routing decisions. Liu and Dessouky (2017) propose a big-M formulation with a two-phase decomposition procedure for the joint passenger and freight train scheduling problem. The first phase is a passenger and freight train scheduling problem, while the second phase deals with the routing optimization for freight trains.

From the above discussion, the big-M formulation is a suitable approach for solving train routing and scheduling problems in complex railway networks. This type of formulation is usually combined with problem decomposition techniques when dealing with large and dense networks. One of the mostly adopted technique deals with the following two steps: 1) the selection of promising train routes; 2) the computation of optimal train schedules for the given routes.

The TT problem, as a core planning in railway industry, has been integrated with several problems, including passenger delay management (Corman et al. 2017), energy-efficient strategies (Wang et al. 2015, Yin et al. 2016), headway train stop plan (Yang et al. 2016, Li et al. 2017), rolling stock circulation and maintenance planning (Giacco et al. 2014, Lai et al. 2015). All these integrated approaches highlight the importance of dealing with integration issues.

In the recent years, the railway track maintenance task scheduling problem has been carefully investigated. In a recent survey, Lidén (2015) points out that the optimizing of train scheduling should be considered simultaneously with the management of track maintenance tasks in order to get better quality solutions in terms of capacity utilization and cost minimization. Budai et al. (2006) focus on the optimization of preventive maintenance activities by scheduling maintenance jobs with maximization of cost saving. Peng et al. (2013) propose an approach based on time-space network to solve the maintenance team scheduling problem with many hard constraints.

The integration of TT and TMTS has attracted an increasing attention in the most recent literature. Vansteenwegen et al. (2016) propose train re-routing and re-timing methods to improve the management of traffic flows with constraints on railway infrastructure availability due to a given plan of maintenance works. As for the first place in 2016 PSC, Carpov et al. (2016) separate the train routing selection process from the joint TT-TMTS optimization problem and develop a constraint programming model to optimize the train timetable and the maintenance task schedule iteratively. A main drawback of their approach is a simplified assumption regarding the management of train speed trajectories for the trains passing through the maintained link. Specifically, the 2016 PSC requires the use of different speed trajectories when computing the running time of the first, second, and other trains in the maintenance area, but these constraints are not directly considered in the approach proposed by Carpov et al. (2016). Furthermore, their algorithmic framework requires, on average, around 2 hours to get their best solution. As for the second place in 2016 PSC, Wang et al. (2016) construct a time-space layered network with decomposition into block sections and identification of minimum running time constraints for each block section. Two heuristics based on the well-known Dijkstra's algorithm are proposed to compute initial schedules, while a mixed integer linear programming model is proposed to improve the solution quality. Regarding the traffic flows on the maintained link, their model is based on the following simplification rules: trains are either not allowed to pass through the affected area for a given time window, or any train passing through this area must adopt a given low-speed profile. Furthermore, the proposed model and algorithm requires up to more than one hour of computation time for some instances provided by 2016 PSC.

Luan et al. (2017) propose an integrated optimization model based on a time-space network for the joint problem of scheduling trains in a railway network and planning preventive railway maintenance in time slots. A Lagrangian relaxation algorithm is proposed and the joint problem is decomposed into train-based shortest path sub-problems. The computational experiments are based on a realistic case study, which is a limited part of the 2016 PSC network. The results show that the integrated optimization method is a good approach to improve the overall performance. In addition, they mention that different reformulations and methods should be investigated to reduce the reported optimality gap.

As the most recent work in the literature, D'Ariano et al. (2017) propose a cell-based mixed integer linear programming model to investigate the trade-off relationship between minimizing the deviation of train schedules and maximizing the number of adjacent maintenance tasks. They also consider the stochastic variability of train travel time and maintenance task duration. Their computational results on a network with a small number of trains show that flexible train routing optimization can often achieve good quality results under stochastic disturbances.

Table 1 presents a detailed comparison of our paper with recent publications related to the integration of TT and TMTS. Even if our work does not deal with a bi-objective optimization, as e.g. in D'Ariano et al. (2017), we consider additional practical constraints related to track maintenance task scheduling. Specifically, we enforce overlapping or contiguous time windows between adjacent maintenance tasks to facilitate the work of maintenance teams and, indirectly, to save maintenance costs. Furthermore, we investigate a more microscopic and precise representation of the railway network, maintenance constraints and traffic flows compared to D'Ariano et al. (2017) and Luan et al. (2017). Finally, we aim to solve significantly more complex realistic instances to near-optimality.

Table 1 Comparison of recent works related to the integration of TT and TMTS

Publication	Objective Function	Maintenance constraints	Modeling approach	Solution algorithm	Solution precision	Largest problem solved by the approach
Luan et al. (2017)	Min total arrival time deviations of all trains	Operational time window; Duration time; Safety	Time-space network	Lagrangian relaxation	1 minute	31 trains; A rail network with 454 nodes, 513 cells, and 4 preventative maintenance time slots
D'Ariano et al. (2017)	Min total arrival time deviations of all trains; Max the number of paired maintenance works	Operational time window; Duration time; Safety; Pairing of maintenance works	Big-*M* formulation	Flexible and fixed train routing strategies; CPLEX solver	1 minute	20 trains; A rail network with 84 nodes, 81 cells, and 6 maintenance cells
This paper	Min total running and dwell times of all trains	Operational time window; Duration time; Safety; Maintenance task adjacency; Train speed restriction; Train speed reduction	Big-*M* formulation	Iterative algorithm with speed up strategies; Gurobi solver	1 second	38 trains; A rail network with 27 stations, 55 segments, 1619 nodes, 1027 cells and 13 maintenance cells

Overall, recent papers focus on various aspects of TT and TMTS, however there is lack of optimization methods for computing integrated solutions. The few existing works on the integration of TT and TMTS lack the ability to handle dense traffic flows on practical-size networks or hard maintenance constraints. The long computation time required to find good quality solutions is also an open issue. All these weak points in the literature motivate the current paper.

3 Problem description

This section describes the integrated problem studied in this paper. We first define the level of detail used in this paper regarding the characteristics of the railway network and the modelling of train traffic flows. We then introduce the problem of integrating TT and TMTS and illustrate the specific constraints to be handle by the mathematical formulation.

3.1 Definition of traffic flows in a microscopic railway network

The railway network infrastructure can be represented at macroscopic or microscopic levels, as illustrated in Fig.1. The TT problem is often managed at a macroscopic level when the network considered is very large and involves huge amount of trains. However, the macroscopic level aggregates several scheduling decisions and cannot be used when there is a need to manage railway operations accurately and to optimize the utilization of complex railway components.

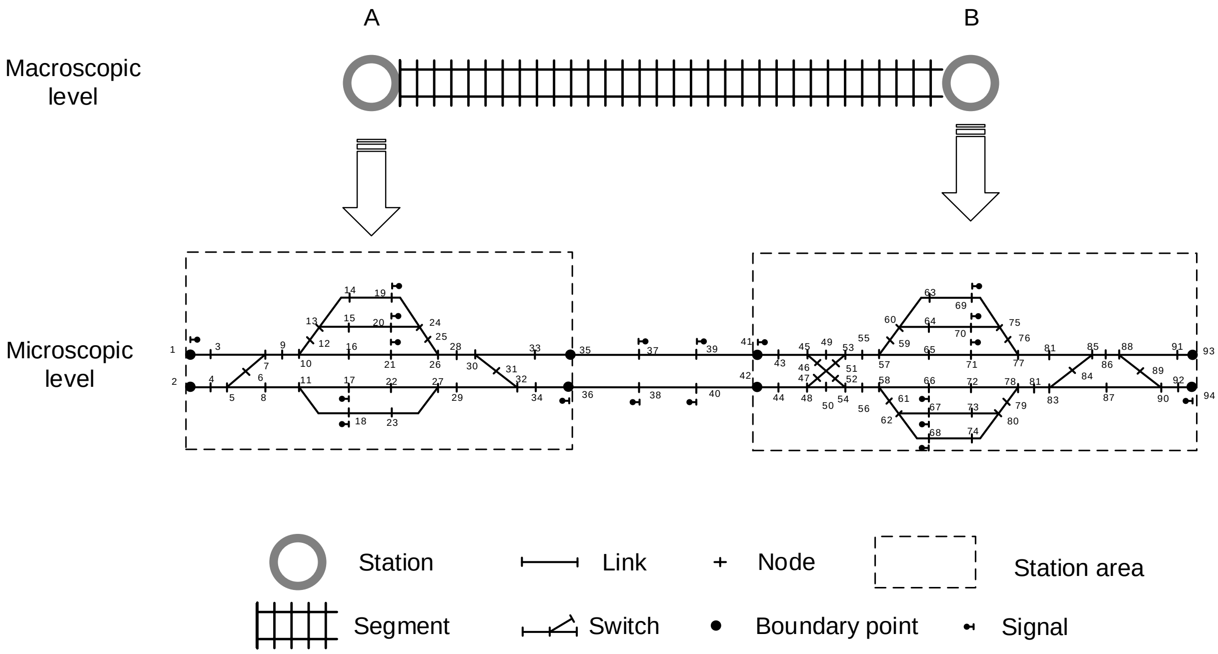


Fig. 1. Illustration of basic railway network components

In the macroscopic level, railway components are abstracted into station areas and track segments between station areas. In a station area, or station, the number of station platforms has clearly an impact on the complexity of the railway infrastructure. A station area can have several platforms to let the trains perform typical station operations, e.g. boarding and alighting of passengers or loading and unloading of goods. Trains can use a station area to perform dwelling, overtaking, crossing or turn-around operations according to their needs. A track segment, or segment, connects two station areas *A* and *B* (see, e.g., Fig. 1) in order to let trains moving from *A* to *B* or vice versa.

In the microscopic level, we need to introduce several additional railway components both in station areas and track segments. A node is a physical point of the network, while a link is a piece of track connecting two adjacent nodes. A switch is a special device of the railway infrastructure that contains at least three links (as shown in Fig 1) and enables trains to be moved from one track to another track. For safety reasons, all the links in one switch shall be occupied or released at the same time. In general, a cell is the atomic resource which consists of a link or switch that allows the passage of one train at a time. It means implicitly that the capacity of a cell is one. The siding cells serve for the trains to stop inside the stations, while the trains cannot have a planned stop on the main tracks as no platform deploys with main track. At microscopic level, station areas are delimited by boundary points that are the nodes identifying the border between a station and a track segment.

A block section consists of a set of cells (links or switches) that are listed in order, and a block section can only be occupied by one train at a time according to the safety regulations. Each block section is delimited by block section nodes. In a signalized fixed block operation, the block section is limited by signals which provide movement authority to enter the block section protected by the signals (Hansen and Pachl, 2014). As the origin or destination nodes of block sections may vary, the block sections are classified into the following four types:

1) An arrival block section is a train route from a home signal to a departure signal within the same station. Home signals control the arrival of trains into stations, while departure signals control the departure of trains from station platforms. In Fig. 2, the train route from node 1 to node 20 is an arrival block section. Since the arrival block sections may end at the siding cells or main track, we define the siding block section nodes and main track block section nodes to distinguish that two kinds of block sections.
2) A departure block section is a train route from a departure signal to a boundary point. In Fig. 2, the train route from node 15 to node 2 is a departure block section.
3) A passing block section is a train route from a boundary point to a passing signal or between two consecutive passing signals in the same track segment. Passing signals are used to control the movement of trains in track segments. In Fig. 2, the train route from node 35 to node 37 is a passing block section.
4) A turn-around block section is a train route from a departure signal to another departure signal in the opposite direction on the same siding cell. This type of block section is required by each train to perform two consecutive services in different traffic directions. For the boarding and alighting of passengers or loading and unloading of goods in a station, each train has a scheduled stop at the end of the assigned arrival block section. In case of a turn-around movement after the scheduled stop, the turn-around train first moves in the turn-around block section and then in the departure block section. In Fig. 2, the train route from node 20 to node 15 is a turn-around block section. The turn-around block section node is defined as the sink node (e.g., node 15 in Fig. 2) of the turn-around block section. We assume that trains cannot perform their scheduled stop at the sink nodes of the turn-around block sections.

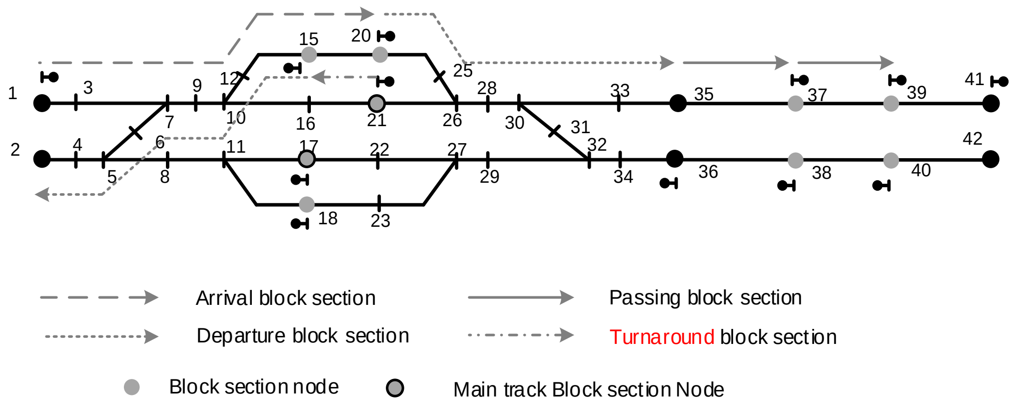


Fig. 2. Illustration of the various types of block section

3.2 Train timetabling

For each train, the TT process needs to define a prescribed, or minimum, dwell time at a siding track of a station area and a prescribed, or minimum, running time on each link of a siding or main track in the network. The TT variables are the block sections used by each train for moving from its origin to its destination, plus its entrance and exit times in each of these block sections. The timing of each train in the network must be defined according to block section occupation constraints, as carefully described in Hansen and Pachl (2014).

A potential conflict between consecutive trains means their blocking times are overlapping. Hence, the blocking time of each block section will be introduced in detailed. Specifically, the cells in each block section are used to check the resource occupation, while the links in each block section are used for the calculation of the running times.

The blocking time consists of reservation time, running time, and release time. The reservation time is composed of the time for clearing the signal, the signal watching time, and the approach time between the signal that provides the approach indication and the signal at the entrance of the block section. The running time begins when the head of the train enters the block section, which is called entrance time, while the running time ends when the tail (the last axle) of the train reaches the end of the block section, which is called exit time. The running time is calculated as the cumulative sum of the travel time on each link which belongs to the block section. We assume that a train can change its traveling direction on the turn-around block sections by switching its head and tail. Thus, its running time on the turn-around block section is computed according to this assumption. The release time is the clearing time of the train length plus the block unlocking time. The minimum running time of each train on the corresponding link is obtained by multiplying its maximum running speed on the link with a given train speed multiplier. The maximum running speed is the free flow speed profile of a train on each link, while the speed multiplier is set according to the throttle setting, the total horsepower of the locomotives, the train tonnage, and other factors associated with each train (INFORMS RAS 2016 PSC). Moreover, to enable a train to absorb small delays, a recovery time is added to the minimum running time of each train. In this paper, the recovery time of each train is associated with every link. This is usually computed as a percentage of the minimum running time, e.g. a typical time supplement in European railway timetables is around 3-7% (Hansen and Pachl, 2014).

Another key element for the train timetabling problem is the train routing optimization that consists to select a route (i.e. a set of block sections) for each train in the microscopic level. The route selected for each train must satisfy all the constraints related to dwelling, overtaking, crossing, and turn-around operations. The origin and destination nodes of each train are connected consecutively by the route selected. In train routing optimization, the combination of different block sections can result in many possible train routes for each train. In the example situation of Fig. 2, a train needs to travel from node 1 to node 39 and has no dwell time requirement at the station. For this train, two alternative routes are identified. The first route consists of arrival block section 1-21, departure block section 21-35, passing block sections 35-37 and 37-39, while the second route includes arrival block section 1-20, departure block section 20-35, passing block sections 35-37 and 37-39. In this paper, train routing is optimized together with train sequencing to achieve the minimum total running and dwell times of all trains. The optimal train sequence between trains depends on the timing and routing decisions taken for each train. Train overtaking is allowed in station areas when the trains are routed through different siding tracks.

3.3 Track maintenance task scheduling

The TMTS process requires to compute the start and end times of each maintenance task, and the maintenance tasks are performed on a given set of cells with a prescribed, or minimum, time duration. An example of the track maintenance task is shown in Fig. 3, where the track maintenance task consists of two maintenance cells. The first maintenance cell includes the maintenance links 3-7, 7-9 and 7-6, while the other maintenance cell contains the maintenance links 8-5, 5-4 and 6-5. Furthermore, each maintenance task must be scheduled in a given time window according to the limited availability of maintenance-related resources. To save track maintenance costs, time windows of maintenance tasks are required to be overlapping or contiguous in case two or more links are adjacent in the railway infrastructure. However, the optimal scheduling of the maintenance tasks clearly depends on the integration between the TT and TMTS processes.

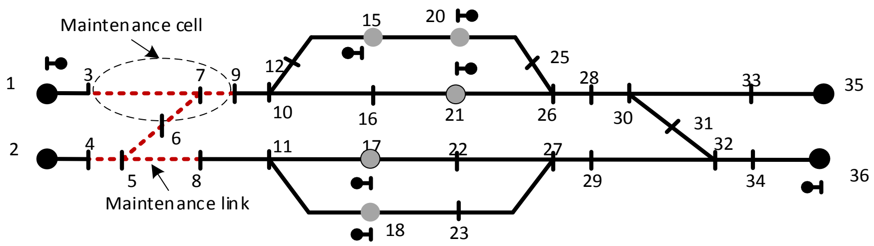


Fig. 3. An illustrative example of track maintenance tasks on two cells

3.4 Integration of train timetabling and track maintenance task scheduling

The Train Timetabling and Track Maintenance Task Scheduling Problem (TTTMTSP) can be defined as follows. The TTTMTSP aims to find an optimally integrated schedule such that all the TT and TMTS constraints are respected. Beyond the constraints of the two individual problems, the TTTMTSP needs to detect and solve conflicting requests of the same cells by train operations and maintenance tasks, since the infrastructure resources can either be occupied by track maintenance or train traffic flows. In the TTTMTSP, train routing and timing decisions need to be coupled with the start and end times of each maintenance task. Specific capacity constraints are required to model occupation and release times of each cell. These constraints are formulated via the identification of safety time intervals between pairs of: consecutive trains, consecutive maintenance tasks, and mixed train operations and maintenance tasks. The resulting optimization problem investigated in this paper is based on the 2016 PSC specifications and requires the minimization of the total running and dwell times of all scheduled trains subject to the TT, TMTS and scheduling integration constraints.

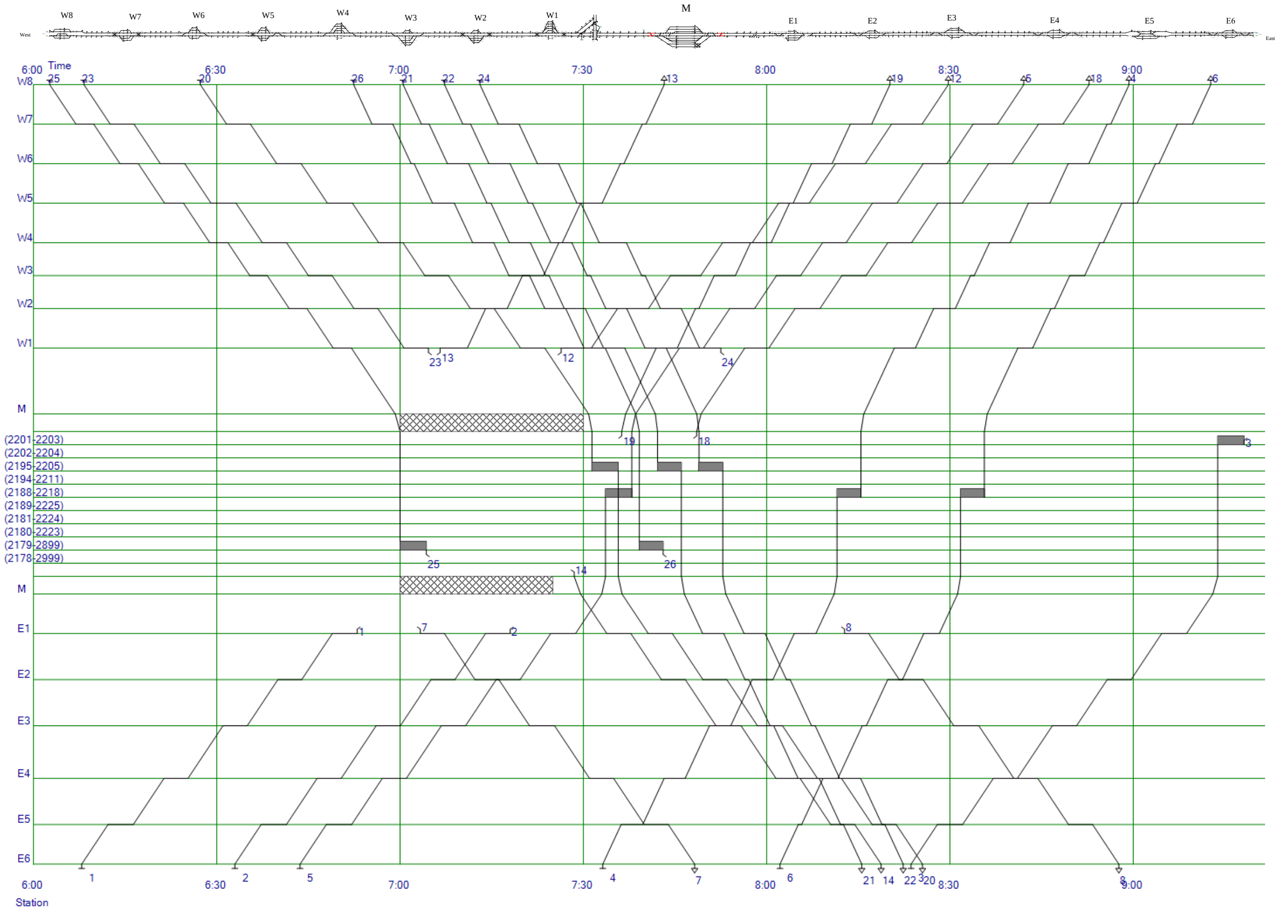


Fig. 4 An TTTMTSP solution for a railway network with 20 trains and two track maintenance tasks

Fig. 5 reports a feasible integration schedule of the TT and TMTS problems. The proposed network is a portion of the 2016 PSC network, including 15 stations (W1, …, W8, M, E1, …, E6) and 3 tactical planning hours. In the proposed TTTMTSP solution, 20 trains are scheduled in the network, including 3 trains from eastern part to western part, 3 trains from western part to eastern part, 6 trains on eastern part only, and 8 trains on western part only. Their routes have the following origin-destination stations: W1-E6, E6-W1, W1-M, M-W1, M-E6 and E6-M. We consider turn-around operations in the largest station area (M) in the network (as shown in Fig. 4 via full grey rectangles). Track maintenance tasks are performed on two cells just outside the largest station area (as shown in Fig. 4 via dotted grey rectangles). In this TTTMTSP solution, all the potential conflicts between trains are solved and no train occupies the two maintained cells while these are maintained by the workers, due to safety reasons. Meanwhile, the routing and timing of the trains and the timing of the track maintenance tasks are implemented such that all the trains can arrive at their destinations as soon as possible.

The TTTMTSP studied in this paper requires to deal with additional practical constraints related to the interaction between the scheduling of track maintenance tasks and the management of train traffic flows. These constraints have been introduced during the 2016 PSC in order to incorporate specific safety requirements identified by practitioners (INFORMS RAS PSC 2016). Specifically, we consider the following two types of additional constraints:

(1) After the completion of a track maintenance task, the related cells are ready to be occupied by the trains, but additional safety requirements are identified for the first two trains scheduled on these cells. The safety requirements deal with the following restricted (maximum) speed profiles: the first (second) train must traverse the maintained cell with a maximum speed of 30 km/h (80 km/h). Fig. 5(*a*) gives an example to illustrate these constraints in which two links (i.e. links 3-7 and 7-9) are maintained and three trains (named trains 1, 2 and 3) must travel on these links after the maintenance task is completed. The first two trains (1 and 2) run according to the constraints on restricted speed profiles, while the third train (3) runs with a faster speed profile. The introduction of this type of additional constraints is to reduce the risk of accidents due to failures related to the implementation of the maintenance tasks.

(2) When dealing with a track maintenance task of a double-track segment, it is possible that the maintained links are related to one traffic direction only. In this situation, the links of this double-track segment related to the other traffic direction are still available for train operations. However, for the safety of workers, all trains must run in these links with a restricted (maximum) speed profile (e.g. with a maximum speed of 50 km/h) during the duration of the entire maintenance task period. Fig. 5(*b*) gives an example of this situation. Link 38-40 is under maintenance, while the links on the opposite traffic direction (i.e. links 35-37, 37-39, and 39-41) are available for train operations. Two trains are scheduled to travel from node 35 to node 41. However, train 1 must run with a slower speed profile than train 2, since the former is scheduled on this track during the track maintenance period, while the latter is scheduled on this track after the completion of all the track maintenance tasks.

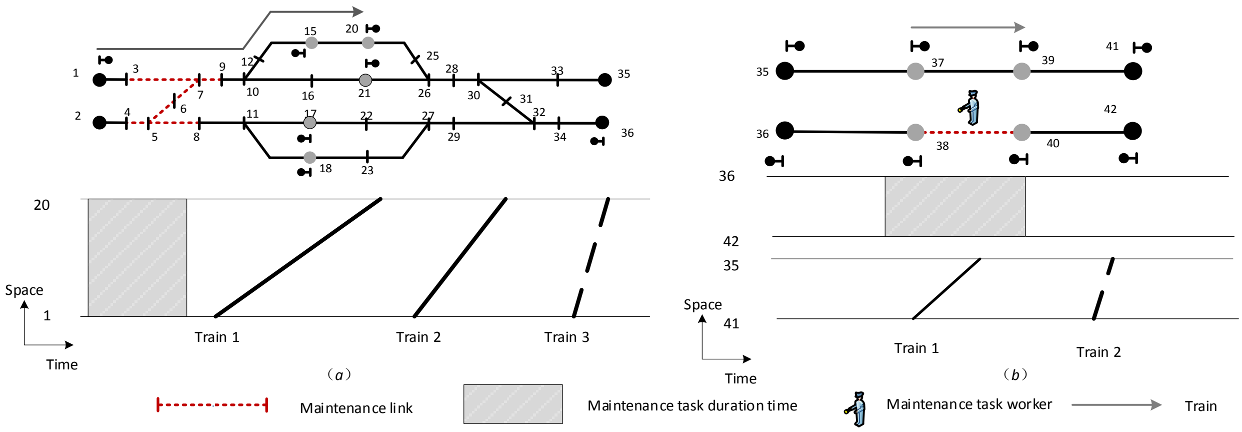


Fig. 5 Train speed profiles on a maintained track (*a*) and on the opposite-direction links of a maintained track (*b*)

4 Mathematical formulation

4.1 Modeling assumptions

Without loss of generality, the following assumptions are made in this paper to facilitate the modeling process.

(1) Objective and constraints related to track maintenance task

It is assumed that the assignment of maintenance resources, such as maintenance teams and maintenance materials, to track maintenance tasks has been determined in advance, as in (Lidén et al., 2015), and we only need to optimize the start and end times of each track maintenance task. Hence, our method has no necessity to consider the maintenance costs. Indeed, all the practical requirements on the track maintenance tasks are modeled as hard constraints for safety reasons. This implies that all constraints related to track maintenance tasks must be satisfied in any TTTMTSP solution.

(2) Set of block sections that one train can travel through

The route that connects the origin and destination nodes of each train is made up of a set of successive block sections. In a TTTMTSP solution, we must select exactly one route for each train among the set of available routes. Besides, we can obtain the set of passing stations according to the origin station, destination station and stop plan related to each train. We can further determine the set of block section nodes on each train route, where those block section nodes are connected by a set of block sections. The set of block sections for each train are thus found based on the selected route.

(3) Sequence of the trains within the track segments

It is assumed that there are no additional devices in the track segments to enable the overtaking between trains, so the sequence of trains within the segments cannot be changed in this paper. The sequence of trains can change when two or more trains use different routes in stations and the involved trains stop in different siding tracks.

(4) Big-M values

The values of big-M in our model are set according to the maximum possible upper bound related to each constraint. Hence, the specified values of big-M may be set differently in the different constraints, without affecting solution quality.

4.2 Definitions

The sets, parameters and decision variables used in this paper are described in the Tables 2, 3 and 4, respectively.

Table 2 Definition of sets

Symbol	Definition
R	Set of trains, index by r, i.e., $r \in R$
S	Set of stations, index by s, i.e., $s \in S$
E	Set of segments, index by e, i.e., $e \in E$.
B	Set of block sections, index by b, i.e., $b \in B$. Block sections can be divided into four types, i.e., arrival block section, departure block section, passing block section and turn-around block sections
C	Set of cells, index by c, i.e., $c \in C$. Cells are the atomic units to identify conflicting block sections
L	Set of links, index by l, i.e., $l \in L$
N	Set of nodes, index by n, i.e., $n \in N$
MOT	Set of track maintenance tasks, index by m, i.e., $m \in MOT$
S_r	Set of stations that train r travels through, $S_r \subset S$
S'_r	Set of stations that train r travels through, but excluding the origin and destination stations, $S'_r \subset S$
B_r	Set of block sections that train r can potentially travel through, $B_r \subset B$
B^a	Set of arrival block sections, $B^a \subset B$
B^p	Set of passing block sections, $B^p \subset B$
$B^+_{r,n}$, $B^-_{r,n}$	Set of block sections that flow out (in) node n and train r can also potentially travel through, $B^+_{r,n}$, $B^-_{r,n} \subset B$
B_c	Set of block sections containing cell c, $B_c \subset B$
B_m	Set of passing block sections that are affected by the track maintenance task m, $B_m \subset B$
C_b	Set of cells in block section b, $C_b \subset C$
C_m	Set of cells included in track maintenance task m, $C_m \subset C$
L_b	Set of links for block section b, $L_b \subset L$
L_c	Set of links for cell c, i.e., $L_c \subset L$
N^{st}	Set of block section nodes within the station, $N^{st} \subset N$

N^m	Set of main track block section nodes, $N^m \subset N$
N^b	Set of the block section nodes that serve as the boundary points, $N^b \subset N$
N_r	Set of nodes that train r can potentially travel through, the origin and destination nodes are excluded, $N^r \subset N$
N_r^{ta}	Set of turn-around block section nodes that train r can travel through, $N_r^{ta} \subset N$
N_s	Set of the block section nodes within station s, $N_s \subset N$
MOT_m	Set of track maintenance tasks adjacent to track maintenance task m, $MOT_m \subset MOT$

Table 3 Definition of parameters

Symbol	Definition
n_r^o, n_r^d	Index of the origin and destination nodes for train r
$[t_r^s, t_r^e]$	Departure time window of train r at the origin node n_r^o
$[mot_m^s, mot_m^e]$	Start time window of track maintenance task m
d_m	Minimum duration time of track maintenance task m
s^n	Index of the station that contains node n
s_r^o, s_r^d	Index of the origin and destination stations for train r
l_b^{last}	The corresponding link of the last cell for an arrival block section $b \in B^a$
w_l	Length of link l (in the unit of mile)
n_b^+	Start node of block section b
n_b^-	End node of block section b
$t_{r,l}$	Minimum running time of train r on link l with the maximum allowed speed profile
$t_{r,l}^{reco}$	Recovery time of train r on link l
$\varepsilon_{b,b'}$	0-1 relationship parameter, equal to 1 if block sections b and b' have cells in common but do not have the same last cell, 0 otherwise
$t_{r,s}^{dwell}$	Minimum dwell time of train r in station s
$v_{l,r}$	Maximum allowed speed of train r on link l
v_m	Maximum allowed speed on the other whole track during operation time of track maintenance task m
v_c^1, v_c^2	Maximum allowed speed of the first (second) train traveling through cell c after track maintenance task on cell c has been completed
$t_{r,b}^{res}$	Reservation time of train r on block section b
$t_{r,b}^{rel}$	Release time of train r on block section b
M	A sufficiently large number that is set specifically for each type of constraint

Table 4 Definition of decision variables

Symbol	Definition
$y_{r,b}^{entr}$, $y_{r,b}^{exit}$	Entrance and exit times of train r on block section b
$x_{r,b}$	0-1 route variable, equal to 1 if train r uses block section b, 0 otherwise
$\mu_{r,b,r',b'}$	0-1 sequence variable, equal to 1 if train r is scheduled earlier on block section b than train r' on block sections b' that is conflicting with block section b, 0 otherwise
$t_{r,s}^{stop}$	Actual dwell time of train r in station s
t_m^{start}	Start time of track maintenance task m
t_m^{end}	End time of track maintenance task m
$\alpha_{r.b,c}$	0-1 entrance time indicator variable, equal to 1 if the entrance time of train r at block section b is larger than or equal to the end time of track maintenance task on cell c, 0 otherwise
$\beta_{r.b,m}$	0-1 exit time indicator variable, equal to 1 if train r exits block section b at its maximum allowed

	speed profile before track maintenance task m starts, 0 otherwise
$z_{r,c}$	0-1 candidate train variable, equal to 1 if train r traverses cell c after the track maintenance task on cell c has been completed, 0 otherwise
$z^1_{r,c}$	0-1 first train variable, equal to 1 if train r is the first train traveling through cell c after the track maintenance task on cell c has been completed, 0 otherwise
$z^2_{r,c}$	0-1 second train variable, equals to 1 if train r is the second train traveling through cell c after the track maintenance task on cell c has been completed, 0 otherwise

4.3 Mathematical model

The mathematical model for Integrated Train Timetabling and Track Maintenance Task Scheduling Problem (ITTTMTSP) is given in the following.

4.3.1 Objective function

The objective of our model is to minimize the total travel time, including the running and dwell times of all trains in the network, as specified in 2016 PSC. Specifically, the objective function is divided into three parts. The first part represents the total running and dwell times of the trains until they release the arrival block sections at their destination stations, the second part represents the sum of actual dwell times of all trains at their destination stations, and the third part represents the sum of running times and recovery times of all trains on their last link of the arrival block sections that connect with their destination nodes.

$$\text{minimize } \mathrm{Z} = \sum_{r\in R}\left(\sum_{b\in B^-_{r,n^d_r}} y^{exit}_{r,b} - \sum_{b\in B^+_{r,n^o_r}} y^{entr}_{r,b}\right) + \sum_{r\in R} t^{stop}_{r,s^d_r} + \sum_{r\in R}\sum_{b\in B^-_{r,n^d_r}} x_{r,b}\left(t_{r,l^{last}_b} + t^{reco}_{r,l^{last}_b}\right) \tag{1}$$

In the investigated railway network and traffic flows, the dwelling and running process of train *r* on the siding cell connecting to its destination node cannot be affected by other trains or track maintenance tasks. Train *r* can thus reach its destination node with the minimum required dwell and running times on this siding cell. The latter dwell and running times correspond to the last two parts of the objective function, that are therefore not affected by the decisions taken during the optimization process. For this reason, the objective function in our set of experiments is simplified by means of expression (2). However, all the computational results presented in this paper will include all the objective function components. The omitted parts in expression (2) will be inserted after fixing all the decision variables in the model.

$$\text{minimize } \mathrm{Z} = \sum_{r\in R}\left(\sum_{b\in B^-_{r,n^d_r}} y^{exit}_{r,b} - \sum_{b\in B^+_{r,n^o_r}} y^{entr}_{r,b}\right) \tag{2}$$

4.3.2 Train movement constraints

(1) Block section usage constraints

The big-*M* method is used to couple the usage of a block section by a train with the entrance and exit times of the train on the corresponding block section. Constraints (3) and (4) imply that the entrance time $y^{entr}_{r,b}$ and exit time $y^{exit}_{r,b}$ of train *r* on block section b will be enforced to be 0 if $x_{r,b}$ is equal to 0, i.e. train r does not occupy block section *b*; otherwise, $y^{entr}_{r,b}$ and $y^{exit}_{r,b}$ are less than or equal to the value of big-*M*.

$$y^{entr}_{r,b} \le M x_{r,b}, \qquad \forall r \in R, b \in B_r \tag{3}$$

$$y^{exit}_{r,b} \le M x_{r,b}, \qquad \forall r \in R, b \in B_r \tag{4}$$

(2) Minimum running time constraints

We recall that the running time of train *r* on block section *b* depends on the different types of block section.

For the departure block sections, passing block sections and turn-around block sections that are not sourcing from the original node of train r, the running time of train r on block section b must be larger than or equal to the sum of the minimum running times and recovery times on all links in block section b. This is enforced by constraints (5).

When train r occupies a departure block section from its origin node, train r needs to perform a dwell time at its origin node within the station. The actual dwell times are enforced by constraints (6).

Constraints (7) are applied to the arrival block sections. Since each arrival block section is released in advance and does not consider the running time on the last link, the minimum running time on the last link is excluded from the second term on the right side of each constraint (7).

$$y_{r,b}^{exit} \geq y_{r,b}^{entr} + x_{r,b} \sum_{l \in L_b} (t_{r,l} + t_{r,l}^{reco}), \qquad \forall r \in R\,, b \in \{b |\, n_b^+ \neq n_r^o, b \in B_r \backslash B^a\} \tag{5}$$

$$\sum_{b \in B_{r,n_r^o}^+} y_{r,b}^{exit} \geq \sum_{b \in B_{r,n_r^o}^+} y_{r,b}^{entr} + \sum_{b \in B_{r,n_r^o}^+} x_{r,b} \sum_{l \in L_b} (t_{r,l} + t_{r,l}^{reco}) + t_{r,s_r^o}^{stop}, \qquad \forall r \in R, n_r^o \notin N^b \tag{6}$$

$$y_{r,b}^{exit} \geq y_{r,b}^{entr} + x_{r,b} \sum_{l \in L_b \backslash l_b^{last}} (t_{r,l} + t_{r,l}^{reco}), \qquad \forall r \in R, b \in B_r \cap B^a \tag{7}$$

(3) Departure time window constraints

Constraints (8) and (9) enforce that the departure time of a train at its origin node should be within a given times window $[t_r^s, t_r^e]$.

$$\sum_{b \in B_{r,n_r^o}^+} y_{r,b}^{entr} \geq t_r^s, \qquad \forall r \in R \tag{8}$$

$$\sum_{b \in B_{r,n_r^o}^+} y_{r,b}^{entr} \leq t_r^e, \qquad \forall r \in R \tag{9}$$

(4) Exit time and entrance time transition between two consecutive block sections

The exit and entrance times on the node of two consecutive block sections are identical if those two block sections are occupied by the same train r. This is required for modelling the continuous train movement on each common block section node. However, on each siding block section node the dwell time is only scheduled when the trains reach the station platform associated with their arrival block section. Specifically, trains are not allowed to perform their dwell time operation at the sink node, i.e. $n \in N_r^{ta}$, of each turn-around block section. Hence, the exit and entrance times on that kind of node included in two consecutive block sections are also identical.

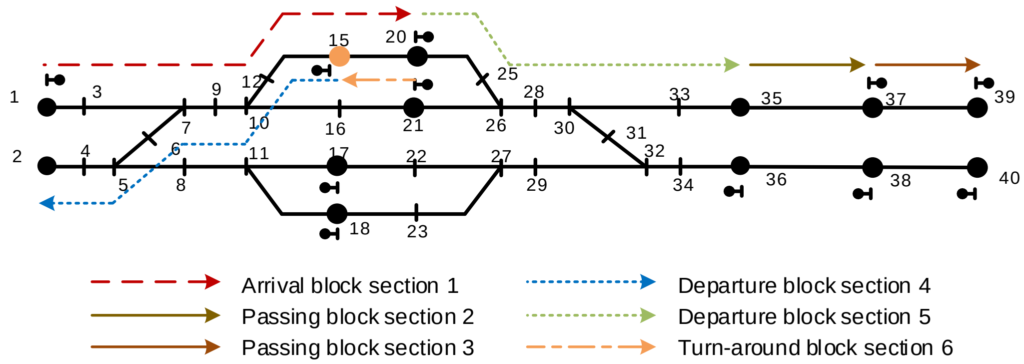


Fig. 6. Examples of entrance time and exit time transition between adjacent block sections

Fig. 6 gives an example situation of a turn-around operation in a station area. First, train r travels on arrival block section 1 until arriving at node 20. Then, train r performs the dwell process at node 20. Soon afterwards, train r changes its traveling direction and travels on turn-around block section 6 and departure block section 4 without any dwell operation. Block section node 15 thus belongs to N_r^{ta} for train r traveling from node 1 to node 20 (arrival block section 1) and from node 20 to node 2 (turn-around block section 6 plus departure block section 4, the exit time from turn-around block section 6 must be equal to the entrance time in block section 4). Concerning the other block sections shown in Fig. 6, the actual dwell times are considered on node 20, that is included in the arrival block section 1 and departure block section 5.

Constraints (10) enforce that the exit and the entrance times on the common node between two consecutive passing block sections or between one turn-around block section and one departure block section are equal. For stopping train *r*, constraint (11) imposes that the transition between the exit time from its arrival block section and the entrance time in its departure block section or turn-around block section is satisfied, including its minimum dwell time of train *r* at the station *s*, its minimum running plus recovery time on the last link associated with the arrival block section. Through constraints (10) and (11), train *r* performs a turn-around operation without dwelling at the end of the turn-around block section. Considering the example of Fig. 6, the transition between the exit time of arrival block section 1 and the entrance time of turn-around block section 6 at node 20 is managed by constraints (11), because node 20 is not defined as a turn-around block section node. On the other hand, the transition between the exit time of turn-around block section 6 and entrance time of departure block section 4 at node 15 is managed by constraints (10), because node 15 is defined as a turn-around block section node. Therefore, train *r* can only dwell at node 20, which is the end of arrival block section 1.

$$\sum_{b'\in B^-_{r,n}} y^{exit}_{r,b'} = \sum_{b\in B^+_{r,n}} y^{entr}_{r,b}, \qquad \forall r\in R, n\in\{n|\, n\neq n^o_r, n\neq n^d_r, n\in (N_r\backslash N^{st})\cup N^{ta}_r\} \tag{10}$$

$$\sum_{n\in N_s\backslash N^{ta}_r}\sum_{b'\in B^-_{r,n}} y^{exit}_{r,b'} + t^{stop}_{r,s} + \sum_{n\in N_s\backslash N^{ta}_r}\sum_{b'\in B^-_{r,n}} x_{r,b'}\left(t_{r,l^{last}_{b'}} + t^{reco}_{r,l^{last}_{b'}}\right) = \sum_{n\in N_s\backslash N^{ta}_r}\sum_{b\in B^+_{r,n}} y^{entr}_{r,b}, \qquad \forall r\in R, s\in S'_r \tag{11}$$

(5) Minimum dwelling time constraints

For train *r* at station *s*, constraint (12) ensures that its actual dwell time $t^{stop}_{r,s}$ is larger than or equal to its minimum dwell time $t^{dwell}_{r,s}$. Obviously, the minimum dwell time is set to 0 if train *r* has not a scheduled stop at station *s*.

$$t^{stop}_{r,s} \geq t^{dwell}_{r,s}, \qquad \forall r\in R, s\in S^r \tag{12}$$

(6) Train stop constraints

Since trains are not allowed to stop on passing block sections on the main tracks, constraints (13) model that if $x_{r,b}$ is equal to 1, train *r* travels through station s^n by using a passing block section *b* and the actual dwell time $t^{stop}_{r,s}$ of train *r* at the station s^n must be equal to 0; otherwise, the actual dwell time $t^{stop}_{r,s}$ is constrained by the big-*M* value.

$$t^{stop}_{r,s} \leq (1-x_{r,b})M, \qquad \forall r\in R, n\in N^m, b\in B^-_{r,n}, s^n\in S^r \tag{13}$$

4.3.3 Train routing constraints

Constraints (14) and (15) are network flow conservation constraints. Each constraint (14) and (15) ensure that train *r* will only choose one route connecting its origin and destination nodes, resulting in a chain of block sections.

$$\sum_{b\in B^+_{r,n}} x_{r,b} - \sum_{b\in B^-_{r,n}} x_{r,b} = \begin{cases} 1 & n=n^o_r \\ -1 & n=n^d_r \\ 0 & otherwise \end{cases}, \qquad \forall r\in R \tag{14}$$

$$\sum_{n\in N^s}\sum_{b'\in B^-_{r,n}} x_{r,b'} = 1, \qquad \forall r\in R, s\in S'_r \tag{15}$$

4.3.4 Block section occupancy constraints

Since the capacity of each block section is one, i.e., each block section can be occupied by at most one train at any time. The sequence of two trains on the conflicting block sections must be determined together with their entrance and exit times on the corresponding block sections, so that all the trains run conflict-free with non-overlapping blocking times.

(1) Two trains choose conflicting block sections

The set of constraints (16)-(20) impose non-overlapping blocking times between two consecutive trains r and r' on two conflicting block sections b and b'. Constraints (16) and (17) address the following block section occupancy situation: two trains conflict on their arrival block sections that share the same last cell ($n \in N^s, b \in B^-_{r,n}, b' \in B^-_{r',n}$). This situation is illustrated in Fig. 7(a) for two trains A and B travelling on their arrival block sections within a station. The routes of trains A and B are different, but those routes are partially overlapping. The overlapping links are the last five links of both train routes, i.e. links 7-9, 9-10, 10-12, 12-15, and 15-20. There is thus a potential conflict between train A and B on these five links. If train A is scheduled earlier than train B on the arrival block section, train B can only start to reserve it after the last link (i.e. link 15-20) is released by train A.

One of the constraints (16) and (17) will be activated if both the variables $x_{r,b}$ and $x_{r',b'}$ are equal to 1. The activation of constraint (16) means train r precedes train r' and thus $\mu_{r,b,r',b'}$ is equal to 1. In this case, the entrance time $y^{entr}_{r',b'}$ of train r' on arrival block section b' must be larger than or equal to the exit time $y^{exit}_{r,b}$ of train r from arrival block section b (including the actual dwell time $t^{stop}_{r,s}$ of train r within station s, the minimum running time t_{r,l^{last}_b} of train r and the recovery time t^{reco}_{r,l^{last}_b} on the last link l^{last}_b of the occupied arrival block section b) plus the time $t^{rel}_{r,b}$ required by train r to release block section b and the reservation time $t^{res}_{r',b'}$ required by train r' to start the occupation of block section b'. Constraints (17) have a similar function of constraints (16) except that train r' precedes train r. It is worth noting that when two trains conflict on their arrival block sections that share the same last cell and this is a main track cell, $t^{stop}_{r,s}$ is enforced to be 0 by the related constraint (13).

$$M(1-x_{r',b'}) + M(1-x_{r,b}) + y^{entr}_{r',b'} - y^{exit}_{r,b} \geq t^{res}_{r',b'} + t^{rel}_{r,b} + t^{stop}_{r,s} + t_{r,l^{last}_b} + t^{reco}_{r,l^{last}_b} - M(1-\mu_{r,b,r',b'})$$
$$,\forall r,r' \in R, n \in N^s, b \in B^-_{r,n}, b' \in B^-_{r',n}, r \neq r', s = s^n, s \in S^r \tag{16}$$

$$M(1-x_{r',b'}) + M(1-x_{r,b}) + y^{entr}_{r,b} - y^{exit}_{r',b'} \geq t^{res}_{r,b} + t^{rel}_{r',b'} + t^{stop}_{r',s} + t_{r',l^{last}_{b'}} + t^{reco}_{r,l^{last}_{b'}} - M\mu_{r,b,r',b'}$$
$$,\forall r,r' \in R, n \in N^s, b \in B^-_{r,n}, b' \in B^-_{r',n}, r \neq r', s = s^n, s \in S^{r'} \tag{17}$$

Each constraint (18) models the situation that train r departs from the origin node n^o_r, which is a block section node within the station ($n^o_r \in N^s$). The set of arrival block sections b', whose end nodes are the same as the corresponding origin node of train r ($b' \in B^-_{r',n^o_r}$), can only be occupied by another train r' after the exit time $y^{exit}_{r,b}$ of train r on the block section plus the release time $t^{rel}_{r,b}$ required by train r and the reservation time required by train r'.

$$M(1-x_{r',b'}) + M(1-x_{r,b}) + + y^{entr}_{r',b'} \geq y^{exit}_{r,b} + t^{res}_{r',b'} + t^{rel}_{r,b}$$
$$,\forall r,r' \in R, b' \in B^-_{r',n^o_r}, n^o_r \in N^s, r \neq r' \tag{18}$$

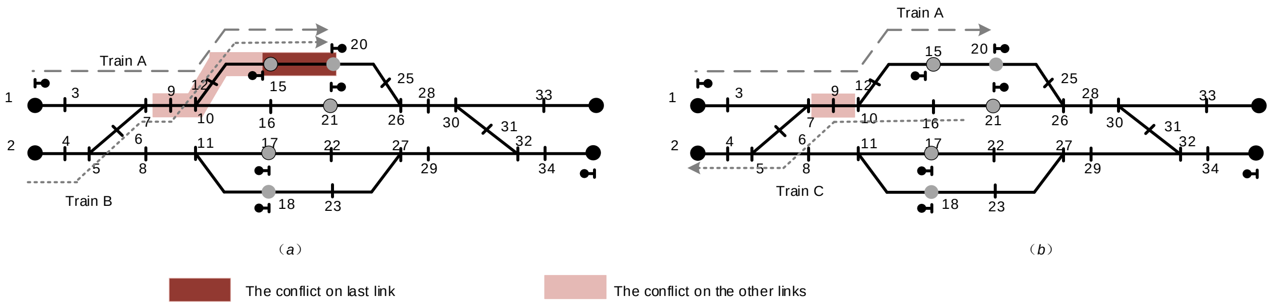


Fig. 7. Examples of potential conflict on the last link (a) and on the other links (b)

Constraints (19) and (20) deal with another situation where the two block sections are conflicting on links different from the last link. These two block sections can be two arrival block sections with different end nodes, two departure block sections, one arrival block section and one departure section, as well as two passing block sections.

Fig. 7(b) gives as an example with two trains A and C in the same network of Fig. 7(a). Train A uses the same arrival block section of Fig. 7(a), while train C runs on its departure block section. The two routes partially overlap on two intermediate links (i.e. 7-9 and 9-10). In this case, train A (or train C) can move on its corresponding block sections only after the conflicting block section is released, so that the blocking times on the conflicting block sections will not overlap.

Each constraint (19) is activated when the variables $x_{r',b'}$, $x_{r,b}$ and $\mu_{r,b,r',b'}$ are both equal to 1, i.e. when train r is scheduled before train r'. The entrance time $y^{entr}_{r',b'}$ of train r' on block section b' must be equal to or larger than the exit time $y^{exit}_{r,b}$ of train r from block section b plus the release time $t^{rel}_{r,b}$ required by train r and the reservation time $t^{res}_{r',b'}$ required by train r'. Constraints (20) have a similar structure of constraints (19) except that $\mu_{r,b,r',b'}$ is equal to 0 and thus train r' is scheduled before train r.

$$M(1-x_{r',b'}) + M(1-x_{r,b}) + y^{entr}_{r',b'} - y^{exit}_{r,b} \geq t^{res}_{r',b'} + t^{rel}_{r,b} - M(1-\mu_{r,b,r',b'}) \\ ,\forall r,r' \in R, b \in B_r, b' \in B_{r'}, r \neq r', \varepsilon_{b,b'} = 1 \tag{19}$$

$$M(1-x_{r',b'}) + M(1-x_{r,b}) + y^{entr}_{r,b} - y^{exit}_{r',b'} \geq t^{res}_{r,b} + t^{rel}_{r',b'} - M\mu_{r,b,r',b'} \\ ,\forall r,r' \in R, b \in B_r, b' \in B_{r'}, r \neq r', \varepsilon_{b,b'} = 1 \tag{20}$$

(2) Sequencing of trains on passing block sections

In our network, the train sequence cannot be altered on passing block sections. We thus set the value of $\mu_{r,b,r',b}$ be equal to the value $\mu_{r,b',r',b'}$ in constraints (21), where b' is the next passing block section adjacent to block section b.

$$\mu_{r,b,r',b} = \mu_{r,b',r',b'}, \qquad \forall r,r' \in R, i \in I, \forall b, b' \in (B_r \cap B_{r'} \cap B^p), r \neq r', b' = b+1 \tag{21}$$

4.3.5 Maintenance task scheduling constraints

(1) Maintenance tasks time constraints

Constraints (22) and (23) enforce that each track maintenance task m must start within a time window $[mot^s_m, mot^e_m]$. Constraints (24) model that the duration time of a track maintenance task should be no less than the required minimum duration time d_m.

$$t^{start}_m \geq mot^s_m, \qquad \forall m \in MOT \tag{22}$$

$$t^{start}_m \leq mot^e_m, \qquad \forall m \in MOT \tag{23}$$

$$t^{end}_m - t^{start}_m \geq d_m, \qquad \forall m \in MOT \tag{24}$$

(2) Maintenance task entrance constraints

The maintenance of the railway tracks requires coupling constraints with the management of the train movements on the infrastructure resources. To guarantee the safety of maintenance teams who should perform the tasks during the maintenance time window, the cells in the track maintenance tasks are unavailable to the trains. Each constraint (25) specifies that if $x_{r,b}$ is equal to 1 and $\alpha_{r,b,c}$ is equal to 0, train r is scheduled before maintenance task m on block section b. In this situation, the exit time $y^{exit}_{r,b}$ of train r on block section b must be less than or equal to the start time t^{start}_m of the maintenance task m. Each constraint (26) formulates the following situation: if both the variables $x_{r,b}$ and $\alpha_{r,b,c}$ are equal to 1, train r is scheduled after maintenance task m on block section b. In this situation, the entrance time $y^{entr}_{r,b}$ of train r on block section b must be larger than or equal to the end time t^{end}_m of maintenance task m.

$$y^{exit}_{r,b} \leq t^{start}_m + M(1-x_{r,b}) + M\alpha_{r,b,c}, \qquad \forall r \in R, m \in MOT, c \in C_m, b \in B_r \cap B_c \tag{25}$$

$$M(1-x_{r,b}) + y^{entr}_{r,b} \geq t^{end}_m - M(1-\alpha_{r,b,c}), \qquad \forall r \in R, m \in MOT, c \in C_m, b \in B_r \cap B_c \tag{26}$$

(3) Maintenance task adjacency constraints

Constraints (27) and (28) are defined to ensure that the time window between adjacent track maintenance tasks is overlapping or contiguous to facilitate the work of maintenance teams and, indirectly, to save maintenance costs.

$$t^{end}_{m_2} \geq t^{start}_{m_1}, \qquad \forall m_1 \in MOT, m_2 \in MOT'_{m_1} \tag{27}$$

$$t^{end}_{m_1} \geq t^{start}_{m_2}, \qquad \forall m_1 \in MOT, m_2 \in MOT'_{m_1} \tag{28}$$

4.3.6 Train speed restriction constraints due to track maintenance

The infrastructure resources affected by maintenance tasks cause the following restrictions to the train traffic flows.

After the track maintenance tasks have been completed on an infrastructure resource, the speed of the first and second trains running on this resource should be restricted for safety reasons. We model this situation via the following two steps. As the first step, we identify the first and second trains passing through each maintenance cell that is involved to track maintenance tasks. As the second step, the maximum speed of the first and second trains traveling through each maintenance cell is restricted according to the given train speed restriction limits.

(1) Identification of the first and second trains after maintenance

With constraints (29), if $\sum_{b\in B_r\cap B_c} x_{r,b}$ is equal to 0, train r will not occupy any block section that contains maintenance cells. In other words, if $z_{r,c}$ is enforced to be 0, train r will not pass through those maintenance cells after the corresponding track maintenance tasks have been completed.

$$z_{r,c} \leq \sum_{b\in B_r\cap B_c} x_{r,b}\,, \qquad \forall r\in R, m\in MOT, c\in C_m \tag{29}$$

Each constraint (30) (each constraint (31)) models the interaction between the entrance time (the exit time) of train r on the block sections containing maintenance cell c and the end time (the start time) of track maintenance task m on cell c. If $z_{r,c}$ is equal to 1, constraint (30) is used to ensure that train r can only occupy the block sections containing cell c after track maintenance task m on cell c has been executed; otherwise, constraint (31) enforces that train r must exit from the block sections containing cell c before the start of track maintenance task m on cell c.

$$\sum_{b\in B_r\cap B_c} y_{r,b}^{entr} + M(1-z_{r,c}) \geq t_m^{end}, \qquad \forall r\in R, m\in MOT, c\in C_m \tag{30}$$

$$\sum_{b\in B_r\cap B_c} y_{r,b}^{exit} \leq t_m^{start} + Mz_{r,c}, \qquad \forall r\in R, m\in MOT, c\in C_m \tag{31}$$

With each constraint (32), if $z_{r,c}^1$ is equal to 1, train r is the first train that travels through maintenance cell c after the corresponding track maintenance task on cell c has been executed. In this case, $z_{r,c}$ must also be equal to 1. Similarly, if $\sum_{r\in R} z_{r,c}$ is larger than or equal to 1, there exists at least one train that travels through maintenance cell c after the track maintenance task on cell c has been executed. We also introduce constraints (33) to ensure that there must be a "first" train that travels through maintenance cell c.

$$z_{r,c}^1 \leq z_{r,c}, \qquad \forall r\in R, m\in MOT, c\in C_m \tag{32}$$

$$\sum_{r\in R} z_{r,c} \leq M\sum_{r\in R} z_{r,c}^1\,, \qquad \forall m\in MOT, c\in C_m \tag{33}$$

Constraints (34) and (35) have a similar structure of constraints (32) and (33) respectively, except that they deal with the existence of the "second" train that travels through maintenance cell c. In addition, the left side of each constraint (35) denotes that there should be at least two trains traveling through maintenance cell c after the track maintenance task on cell c has been executed. Once the latter condition is verified, constraint (35) enforces that a "second" train must travel through maintenance cell c.

$$z_{r,c}^2 \leq z_{r,c}, \qquad \forall r\in R, m\in MOT, c\in C_m \tag{34}$$

$$\sum_{r\in R} z_{r,c} - 1 \leq M\sum_{r\in R} z_{r,c}^2\,, \qquad \forall m\in MOT, c\in C_m \tag{35}$$

We define constraints (36) to enforce that, after the maintenance task on a maintenance cell has been executed, the entrance time of the first train must be less than or equal to the entrance time of the other trains on each block section that is related to this maintenance cell. For a cell c, constraint (36) is activated if each of the variables $z_{r,c}^1$, $z_{r,c}$ and $z_{r',c}$ is equal to 1. This implies the following condition: after the maintenance task on cell c has been executed, train r is the first train that travels through maintenance cell c and train r' is one of the other trains that also travels through cell c. With this condition, the entrance time $\sum_{b\in B_r\cap B_c} y_{r,b}^{entr}$ of train r on the block section that contains maintenance cell c

must be less than or equal to the entrance time $\sum_{b\in B_{r'}\cap B_c} y_{r',b}^{entr}$ of train r'.

$$M\left(1-z_{r,c}^1\right)+M\left(1-z_{r,c}\right)+M\left(1-z_{r',c}\right)+\sum_{b\in B_{r'}\cap B_c} y_{r',b}^{entr} \geq \sum_{b\in B_r\cap B_c} y_{r,b}^{entr}, \forall r,r'\in R, m\in MOT, c\in C_m, r\neq r' \tag{36}$$

Constraints (37) model additional constraints on the train traffic flows after the track maintenance task on cell c has been executed. These constraints deal with the entrance time of the second train and the entrance time of the other trains on the block sections that contain maintenance cell c. For a cell c, constraint (37) is activated if each of the variables $z_{r,c}^2$, $z_{r,c}$ and $z_{r',c}$ is equal to 1 and the variable $z_{r',c}^1$ is equal to 0. This case means that, after the maintenance task on cell c has been executed, train r is the second train that travels through maintenance cell c and train r' is one of the other trains (but not the first train) that travels through maintenance cell c. To model the sequencing constraint between train r and the following trains, the entrance time $\sum_{b\in B_r\cap B_c} y_{r,b}^{entr}$ of train r on the block section that contains maintenance cell c must be less than or equal to the entrance time $\sum_{b\in B_{r'}\cap B_c} y_{r',b}^{entr}$ of train r'.

$$M\left(1-z_{r,c}^2\right)+Mz_{r',c}^1+M\left(1-z_{r,c}\right)+M\left(1-z_{r',c}\right)+\sum_{b\in B_{r'}\cap B_c} y_{r',b}^{entr} \geq \sum_{b\in B_r\cap B_c} y_{r,b}^{entr} \tag{37}$$

$$, \forall r,r'\in R, m\in MOT, c\in C_m, r\neq r'$$

Each constraint (38) describes the following sequencing constraint: train r can only be either the first train or the second train that travels through maintenance cell c after the track maintenance task on cell c has been executed.

$$z_{r,c}^1+z_{r,c}^2\leq 1, \qquad \forall r\in R, m\in MOT, c\in C_m \tag{38}$$

(2) Restricting the maximum speed for the first and second trains after maintenance

The following set of constraints enforce a restricted speed profile for the first and second trains on each block section that is subject to track maintenance tasks. Since there are various types of block sections, we need to consider two cases of speed restriction. The first case is valid for all types of block sections but the arrival block sections, while the second case is dedicated to the arrival block sections.

To model the two cases of speed restriction, we need the following definitions. Let $L_b^{res}=\{l|\, l\in L_b\cap L_c, c\in C_b\cap C_m\}$ denote the set of links that are contained both in block section b and in maintenance cell c, and let $L_b^{unres}=\cup\{l|\, l\in L_b\backslash L_c, c\in C_m\}$ represent the set of links that are contained in block section b but are not contained in any maintenance cell $c\in C_m$. By adopting the definitions of $L_b^{res}$and L_b^{unres}, the maximum speed of the first and second trains on the links belonging to the set L_b^{res} will be restricted, while the maximum speed of the first and second trains on the links belonging to the set L_b^{unres} will not be affected.

For the first case of speed restriction, the minimum running times of the first and second trains on the maintained block sections are enforced by constraints (39). Specifically, each constraint (39) handles the following situation: train r is either the first or second train traveling through the maintenance cells that are contained in block section b. With constraint (39), the restricted minimum block section running time $t_{r,b}^{restrict}$ is thus obtained by computing the restrained speed profile of train r on block section b. This speed profile is obtained by considering the minimum value between the speed limit of train r and the link speed limit plus a given train speed multiplier.

For the second case of speed restriction, the last link of the arrival block sections needs to be excluded. Therefore, we define two new sets of links $L_b^{res,a}=\{l|\, l\in L_b\cap L_c, l\neq l_b^{last}, c\in C_b\cap C_m\}$ and $L_b^{unres,a}=\{l|\, l\in L_b\backslash L_c, l\neq l_b^{last}, c\in C_m\}$. After defining these new sets, constraints (40) are used to compute the minimum running times of the first and second trains on the maintained arrival block sections. The running times computed by constraints (39) and (40) also include the possible recovery time of each train r in each link l.

$$t_{r,b}^{restrict} = \sum_{c\in C_b\cap C_m}\left(z_{r,c}^1 \sum_{l\in L_b^{res}}\left(\lceil w_l/\min\{v_{l,r}, v_c^1\}\rceil + t_{r,l}^{buf}\right) + z_{r,c}^2 \sum_{l\in L_b^{res}}\left(\lceil w_l/\min\{v_{l,r}, v_c^2\}\rceil + t_{r,l}^{reco}\right)\right) + \sum_{l\in L_b^{unres}}\left(t_{r,l} + t_{r,l}^{reco}\right) \quad \forall\, b \in B_r \backslash B^a \tag{39}$$

$$t_{r,b}^{restrict} = \sum_{c\in C_b\cap C_m}\left(z_{r,c}^1 \sum_{l\in L_b^{res,a}}\left(\lceil w_l/\min\{v_{l,r}, v_c^1\}\rceil + t_{r,l}^{buf}\right) + z_{r,c}^2 \sum_{l\in L_b^{res,a}}\left(\lceil w_l/\min\{v_{l,r}, v_c^2\}\rceil + t_{r,l}^{reco}\right)\right) + \sum_{l\in L_{b,c}^{unres,a}}\left(t_{r,l} + t_{r,l}^{reco}\right) \quad \forall\, b \in B^a \tag{40}$$

After obtaining the minimum running time of the first and second trains on block section b, the entrance and exit times of any other train r on block section b is restricted by constraints (41). Specifically, each constraint (41) specifies the following condition. If the variable $x_{r,b}$ is equal to 1, the difference between the exit time $y_{r,b}^{exit}$ of train r from block section b and the entrance time $y_{r,b}^{entr}$ of train r on block section b must be larger than or equal to the restricted minimum block section running time $t_{r,b}^{restrict}$.

$$M(1 - x_{r,b}) + y_{r,b}^{exit} \geq y_{r,b}^{entr} + t_{r,b}^{restrict}, \qquad \forall r \in R, m \in MOT, b \in B_r, C_b \cap C_m \neq \emptyset \tag{41}$$

4.3.7 Train speed reduction constraints during maintenance

The following set of constraints reduce the speed of all trains travelling in a track during the execution of the track maintenance tasks by the maintenance team in the opposite travelling direction. In a double-track segment, the speed profile of the trains traversing the track needs to be restrained (on the whole track) during the overall time window in which an opposite-direction cell of this track is being maintained. Constraints (42)-(46) are used to determine whether the running time of train r on block section b will be affected by a track maintenance task on maintenance cell c. Constraints (42) and (43) indicate whether train r enters the block section b after the track maintenance task on cell c has been executed. Constraint (42) states that if the variable $\alpha_{r,b,c}$ is equal to 1, the entrance time $y_{r,b}^{entr}$ of train r on block section b is larger than the completion time t_m^{end} of maintenance task m. In other words, train r can occupy maintenance cell c by using block section b after maintenance task m has been executed. Constraint (43) states that if the variable $\alpha_{r,b,c}$ is equal to 0, the entrance time $y_{r,b}^{entr}$ of train r on block section b is less than the completion time t_m^{end} of maintenance task m.

$$y_{r,b}^{entr} + M(1 - \alpha_{r,b,c}) \geq t_m^{end}, \qquad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_m) \tag{42}$$

$$y_{r,b}^{entr} \leq t_m^{end} + M\alpha_{r,b,c}, \qquad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_m) \tag{43}$$

Constraints (44) and (45) denote whether train r can travel through block section b at its maximum allowed speed profile before track maintenance tasks on cell c start. In each constraint (44), the variable $\beta_{r,b,c}$ is enforced to be 1 only if the sum of the entrance time $y_{r,b}^{entr}$ of train r on block section b and the minimum running time $\sum_{l\in L_b} t_{r,l}$ of train r on block section b is less than the start time of maintenance task m. This implies that train r can travel through maintenance cell c at its maximum allowed speed profile before maintenance task m on cell c starts. Constraints (45) model the following case: the variable $\beta_{r,b,c}$ is enforced to be 0 only if the sum of the entrance time $y_{r,b}^{entr}$ of train r on block section b and the minimum running time $\sum_{l\in L_b} t_{r,l}$ of train r on block section r is larger than the start time of maintenance task m. This implies that train r can travel through maintenance cell c at its maximum allowed speed profile after maintenance task m on cell c starts.

$$y_{r,b}^{entr} + \sum_{l\in L_b} t_{r,l} + M\beta_{r,b,c} \geq t_m^{start}, \qquad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_m) \tag{44}$$

$$y_{r,b}^{entr} + \sum_{l \in L_b} t_{r,l} \leq t_m^{start} + M(1 - \beta_{r,b,c}), \qquad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_m) \tag{45}$$

Constraints (46) ensure that only one of two conditions described by constraints (44) and (45) may happen, i.e. either train r travels through maintenance cell c after the end time t_m^{end} of maintenance task m or train r travels through maintenance cell c before the start time t_m^{start} of maintenance task m. In each constraint (47), the speed profile of train r on block section b is restrained when the running time window of train r intersects with the maintenance time window on cell c. Constraint (47) is activated if each of the variables $\alpha_{r,b,c}$, $\beta_{r,b,c}$ and $x_{r,b}$ is equal to 0. In this case, the exit time $y_{r,b}^{exit}$ of train r from block section b must be larger than or equal to the sum of the entrance time of train r on block section b plus the minimum running time of train r on block section b. As depicted in constraint (48), the minimum running time of train r on block section b increases to $t_{r,b}^{reduction}$, where v_m is the maximum allowed train speed on the whole track in the opposite direction with respect to track maintenance task m. The minimum running time $t_{r,b}^{reduction}$ also includes the possible recovery time of train r on link l.

$$\alpha_{r,b,c} + \beta_{r,b,c} \leq 1, \qquad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_m) \tag{46}$$

$$M(\alpha_{r,b,c} + \beta_{r,b,c}) + M(1 - x_{r,b}) + y_{r,b}^{exit} \geq y_{r,b}^{entr} + t_{r,b}^{reduction}, \qquad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_m) \tag{47}$$

$$t_{r,b}^{reduction} = \sum_{l \in L_b} \left(\lceil w_l / \min\{v_{l,r}, v_m\} \rceil + t_{r,l}^{buf} \right) \tag{48}$$

4.3.8 Domain of variables

The domain of variables in the model is defined by expressions (49)- (55) and is next summarized. The entrance and exit times of the trains on the block sections, the actual dwell times of the trains at the stations and the start and end times of track maintenance tasks are defined as integer variables. The rest of the variables are defined as binary variables, since these denote the occupancy constraints between two trains and two block sections.

$$y_{r,b}^{entr}, y_{r,b}^{exit} \in N, \qquad \forall r \in R, b \in B_r \tag{49}$$

$$x_{r,b} \in \{0,1\}, \qquad \forall r \in R, b \in B_r \tag{50}$$

$$\mu_{r,b,r',b'} \in \{0,1\}, \qquad \forall r \in R, b \in B_r,\ r' \in R, b \in B_{r'}: r \neq r' \tag{51}$$

$$t_{r,s}^{stop} \in N, \qquad \forall r \in R, s \in S^r \tag{52}$$

$$t_m^{start}, t_m^{end} \in \{0,1\}, \qquad \forall m \in MOT \tag{53}$$

$$\alpha_{r,b,c}, \beta_{r,b,c} \in \{0,1\}, \qquad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_c) \tag{54}$$

$$z_{r,c}, z_{r,c}^1, z_{r,c}^2 \in \{0,1\}, \qquad \forall r \in R, m \in MOT, c \in C_m \tag{55}$$

5 Solution methods

This section deals with the development of solution approaches for the ITTTMTSP, which is modelled as a big-*M* formulation. As we have shown in the previous section, our formulation uses the big-*M* method that is a suitable approach for solving train routing and scheduling problems in complex railway networks. Algorithmic approaches for big-*M* formulations usually decompose the overall problem into two steps: 1) the selection of promising train routes; 2) the computation of optimal train schedules for the given routes (D'Ariano et al. 2008, Corman et al. 2010, Mu and Dessouky 2011, Pellegrini et al. 2014-2015, Samà et al. 2017a-2017c, Liu and Dessouky 2017). Inspired by the related literature, we decompose the ITTTMTSP in train scheduling variables, train routing variables, and track maintenance scheduling variables, and we deal with the coupling relationships between the various types of variables. Based on this idea of decomposition, we present an algorithmic framework with an iterative algorithm for the ITTTMTSP that iteratively solves train scheduling and/or routing problems without or without consideration of maintenance task scheduling. The iterative framework is also motivated by the fact that, when integrating train timetabling and track maintenance task scheduling, the selection of optimal train routes, orders and timing is strongly interrelated with the management of the infrastructure maintenance process and the scheduling of track maintenance tasks.

This section is organized as follows. First, we propose the general structure of the various phase of the proposed iterative algorithm. Second, we introduce the mathematical formulation to be solved in each phase. Third, we present the detailed pseudo-code of the algorithm. We also describe speed-up strategies implemented to improve the solution process.

5.1 Algorithmic framework

As the number of variables and constraints of the ITTTMTSP is huge, our idea is to decompose the studied problem into the following three phases: 1) Train Scheduling and Routing Without Maintenance task (TSRWM) phase; 2) Train Scheduling and Track Maintenance Task Scheduling (TSTMTS) phase; 3) Train Scheduling and Rerouting (TSR) phase. In our approach, each phase is modelled as a mathematical formulation of a simplified ITTTMTSP and is solved by a standard MILP solver. Fig. 8 shows how the various phases are used in our iterative method. The process starts with the collection of railway data about the infrastructure, train traffic flows and maintenance tasks. TSRWM completes the first phase executed by the process. This is only performed once to compute the initial route selected for each train disregarding the maintenance task scheduling constraints of the ITTTMTSP. The other two phases are iteratively used during the solution process, starting from TSTMTS with the initial routes computed by TSRWM. At each iteration, TSTMTS improves the maintenance task start and end times based on the current best route computed by TSR for each train, while TSR improves the train routes based on the maintenance task start and end times computed by TSTMTS. The iterative process terms when a stopping criterion is reached and delivers the best ITTTMTSP solution.

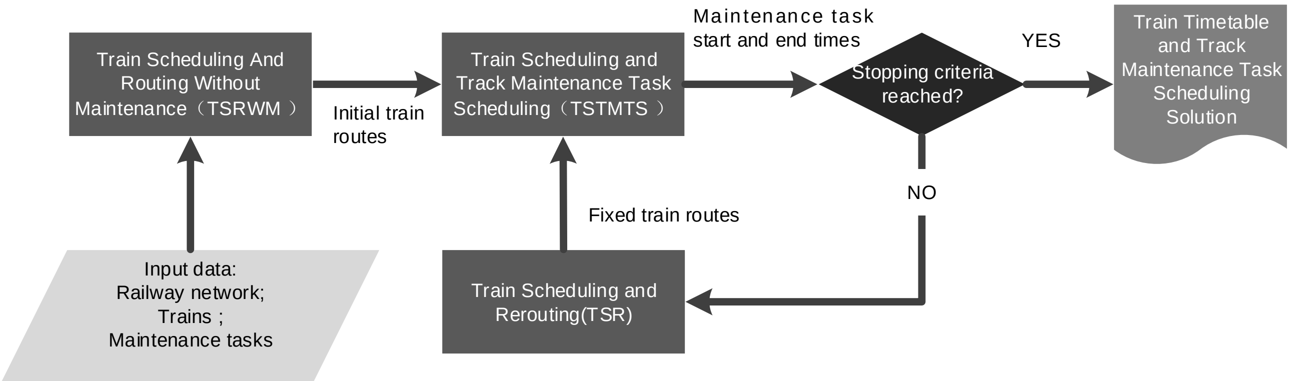


Fig. 8. Algorithmic framework of the solution method

5.2 TSRWM phase

This phase aims to compute an initial route for each train and thus to generate the set of initial routes $B_{r,initial}$. The routes are selected by minimizing the difference between the exit and entrance time of each train in each block section, without considering the maintenance task constraints of the ITTTMTSP. The formulation of TSRWM is the following:

$$\text{TSRWM:} \quad \text{minimize Z} = \sum_{r\in R}\left(\sum_{b\in B^{-}_{r,n^d_r}} y^{exit}_{r,b} - \sum_{b\in B^{+}_{r,n^o_r}} y^{entr}_{r,b}\right)$$

Subject to:

Constraints (3)–(21)

$y^{entry}_{r,b}, y^{exit}_{r,b} \in N, \quad \forall r \in R, \forall b \in B_r$

$x_{r,b} \in \{0,1\}, \quad \forall r \in R, \forall b \in B_r$

$\mu_{r,b,r',b'} \in \{0,1\}, \quad \forall r \in R, \forall b \in B_r, \forall r' \in R, \forall b \in B_{r'}: r \neq r'$

$t^{stop}_{r,s} \in N, \quad \forall r \in R, \forall s \in S^r \cup s^o_r \cup s^d_r$

As shown in the TSRWM formulation, only the set of constraints associated with train timetabling are considered. The resulting problem is a typical TT with train routing and timing variables and minimization of the total travel time.

5.3 TSTMTS phase

This phase is performed to get feasible start and end times for each maintenance task in the ITTTMTSP. To this aim, we need to consider all the ITTTMTSP constraints. However, we consider a given route for each train to get a feasible solution to ITTTMTSP. The train routing information is $B_{r,initial}$ in the first run of the TSTMTS phase, while in the other runs is $B_{r,re-route}$. The latter information is taken from the solution of the TSR phase and is translated into Constraints (56). The detail formulation of TSTMTS is as follows:

TSTMTS: minimize $Z = \sum_{r\in R}\left(\sum_{b\in B^-_{r,n^d_r}} y^{exit}_{r,b} - \sum_{b\in B^+_{r,n^o_r}} y^{entr}_{r,b}\right)$

Subject to:

Constraints (3)–(48)

$x_{r,b} = 1\ \forall r \in R, b \in B_{r,re-route}$ (56)

$y^{entry}_{r,b}, y^{exit}_{r,b} \in N, \quad \forall r \in R, \forall b \in B_r$

$x_{r,b} \in \{0,1\}, \quad \forall r \in R, \forall b \in B_r$

$\mu_{r,b,r',b'} \in \{0,1\}, \quad \forall r \in R, \forall b \in B_r,\ \forall r' \in R, \forall b \in B_{r'}: r \neq r'$

$t^{stop}_{r,s} \in N, \quad \forall r \in R, \forall s \in S^r \cup s^o_r \cup s^d_r$

$t^{start}_m, t^{end}_m \in N, \quad \forall m \in MOT$

$\alpha_{r,b,c}, \beta_{r,b,c} \in \{0,1\}, \quad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_c)$

$z_{r,c}, z^1_{r,c}, z^2_{r,c} \in \{0,1\}, \quad \forall r \in R, \forall m \in MOT, \forall c \in C_m$

The solution of the TSTMTS formulation can be viewed the solution of the ITTTMTSP with pre-defined train routes. We use this solution to get feasible times $\widetilde{t^{start}_m}, \widetilde{t^{end}_m}$ for each maintenance task m.

5.4 TSR phase

This phase is dealing with the computation of optimal train routes and times for the ITTTMTSP without changing the maintenance task scheduling decisions taken by the TSTMST phase. For each maintenance task m, we thus take as input the feasible times $\widetilde{t^{start}_m}, \widetilde{t^{end}_m}$ computed by TSTMST at the end of its last execution. This information is used to generate Constraints (57) and (58), enforcing the given maintenance task times. The TSR formulation is as follows:

TSR: minimize $Z = \sum_{r\in R}\left(\sum_{b\in B^-_{r,n^d_r}} y^{exit}_{r,b} - \sum_{b\in B^+_{r,n^o_r}} y^{entr}_{r,b}\right)$

Subject to:

Constraints (3)–(48)

$t^{start}_m = \widetilde{t^{start}_m}, \forall m \in MOT$ (57)

$t^{end}_m = \widetilde{t^{end}_m}, \forall m \in MOT$ (58)

$y^{entry}_{r,b}, y^{exit}_{r,b} \in N, \quad \forall r \in R, \forall b \in B_r$

$x_{r,b} \in \{0,1\}, \quad \forall r \in R, \forall b \in B_r$

$\mu_{r,b,r',b'} \in \{0,1\}, \quad \forall r \in R, \forall b \in B_r,\ \forall r' \in R, \forall b \in B_{r'}: r \neq r'$

$t^{stop}_{r,s} \in N, \quad \forall r \in R, \forall s \in S^r \cup s^o_r \cup s^d_r$

$t^{start}_m, t^{end}_m \in N, \quad \forall m \in MOT$

$\alpha_{r,b,c}, \beta_{r,b,c} \in \{0,1\}, \quad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_c)$

$z_{r,c}, z^1_{r,c}, z^2_{r,c} \in \{0,1\}, \quad \forall r \in R, \forall m \in MOT, \forall c \in C_m$

The TSR formulation can be considered as a train scheduling and routing problem with the addition of maintenance task constraints. As discussed in Section 2, this problem is difficult to solve in a reasonable computation time for practical-size instances. We therefore propose the following two strategies in the TSR phase to speed up the solution process.

Strategy 1: Defining a promising subset of first and second trains

The idea behind this strategy is to reduce the number of track maintenance task constraints in the TSR formulation. Specifically, we recall that the constraints (29)-(38) in the ITTTMTSP formulation determine the first and second trains after the maintenance task. The number of these constraints increases rapidly with the number of trains. We thus propose the following set $R_{f,m}$ of first and second trains to decrease the number of these constraints:

$$R_{f,m} = \{r \mid \widetilde{t_m^{end}} < \tilde{y}_{r,b}^{entry} < \widetilde{t_m^{end}} + t_m \quad \forall r \in R, m \in MOT, c \in C_m, b \in (B_r \cap B_c)\},$$

in which t_m is a given time window for each maintenance task *m* to select the candidate first and second trains. The times $\tilde{y}_{r,b}^{entry}, \tilde{y}_{r,b}^{exit}$ for each train *r* and block section *b* are taken from the last execution of the TSTMTS phase. Given $R_{f,m}$ for each maintenance task *m*, Constraints (59) are used to decrease the number of constraints (29)-(38).

$$z_{r,c}^{1} = 0, z_{r,c}^{2} = 0 \quad \forall r \notin R_{f,m}, m \in MOT, c \in C_m, b \in (B_r \cap B_c) \tag{59}$$

Strategy 2: Limiting the solving time dedicated to the TSR phase

This strategy aims to balance the time spent between the TSR and TSTMTS phases in each iteration of the solution method. To this aim, we set maximum solving time $t_{TSR,max}$ for each execution of the TSR phase. Here, the trade-off is between computing the optimal train routes and times for a given setting of the maintenance task times and investigation other settings of the integrated optimization method, i.e. performing other iterations of the solution method. We note that TSR is eventually solved with a time limit of computation, since this is often the most time-consuming phase of our iterative algorithm, while the other phases are solved by the MILP solver without a limited computation time.

The potential improvement of using Strategies 1 and 2 will be quantified in Section 6 on several realistic test cases.

5.5 Iterative algorithm

With the approaches proposed to solve each phase of the algorithmic framework of Fig. 8, we can compute the following lower and upper bounds to the optimal ITTTMTSP solution. The optimal solution to the TSRWM phase, i.e. obj_{TSRWM}, corresponds to a lower bound obj_{lower} since this is the optimal TT solution disregarding all the TMTS constraints and thus a relaxed (non-integrated) version of the ITTTMTSP. This solution corresponds to the ideal situation in which TT and TMTS are completely disjoint, i.e. the constraints of the TMTS problem have no impact on the optimal solution of the TT problem. An upper bound obj_{upper} to the optimal ITTTMTSP solution is obtained by solving the TSTMTS (obj_{TSTMTS}) or TSR phase (obj_{TSR}), since each of these phases always delivers a feasible integrated solution.

Based on the proposed lower and upper bounds, we design the following stopping criterion for the algorithm:

$$\frac{obj_{\mathrm{upper}} - obj_{\mathrm{lower}}}{obj_{\mathrm{lower}}} < \varepsilon \tag{60}$$

Expression (60) uses the parameter ε to fix a certain satisfactory ratio on upper and lower bounds, as a criterion to stop the algorithm. Another stopping criterion is a total maximum solving time t_{max} given to the iterative process. The iterative algorithm will be terminated when one of the two stopping criteria is reached.

Fig. 10 provides a pseudo-code of the iterative algorithm proposed in this paper. The algorithm starts with an *initialization* step. First, we model the railway infrastructure, train traffic flows and maintenance tasks in the ITTTMTSP as the proposed big-*M* MILP formulation. Second, we solve the mathematical formulation for TSRWM to get a lower bound and $B_{r,initial}$. Third, we solve the mathematical formulation for TSTMTS with $B_{r,initial}$ to get an upper bound, plus the following values for each maintenance task $\widetilde{t_m^{start}}, \widetilde{t_m^{end}}$ and pair train *r* and block section *b* $\tilde{y}_{r,b}^{entry}, \tilde{y}_{r,b}^{exit}$.

After the initialization step, the algorithm performs the *iterative process* until stopping criteria are not reached. We recall that we adopt two stopping criteria: the search process is halted when a near-optimal solution is found (depending to the value given ε) or when a time limit is exceeded. Specifically, $t_c > t_{max}$ is used at the end of each TSTMTS or TSR phase of the iterative process to check whether the current solving time t_c is larger than the maximum solving time. If the stopping criteria are not reached, a new iteration is performed as follows. The algorithm first solves the TSR phase with Constraints (57)-(59) to get $B_{r,re-route}$, and then solves the TSTMTS phase with Constraints (56), i.e. with $B_{r,re-route}$, to get $\widetilde{t_m^{start}}, \widetilde{t_m^{end}}$ for each maintenance task. The latter values are used to generate Constraints (57)-(58). If Strategy 1 is not performed, Constraints (59) will not be generated.

A maximum solving $t_{phase,max}$ is associated with each phase. In general, this is dynamically used in each iterative phase of the algorithm to fix a current maximum solving time compatible with t_{max}. If Strategy 2 is performed, $t_{phase,max} = \min\{t_{TSR,max}, t_{max} - t_c\}$ in which $t_{TSR,max}$ is the maximum solving time for any TSR phase. During the iterative process, the best upper bound value obj_{upper} and the best solution found are (eventually) updated at the end of each TSTMTS or TSR phase. The output of this algorithm is: t_c, the best solution found and the related information.

Input: Collect strategies, parameters and data, including the sets R, C, L, N, MOT. Current iter *itr* ← 0, upper bound obj_{upper} ← +∞; lower bound obj_{lower} ← -∞, set current total solving time t_c ← 0, set maximum total solving time t_{max}, set maximum TSR solving time $t_{TSR,max}$, set maximum phase solving time $t_{phase,max}$, set current phase solving time t_{phase} ← 0, set t_m for $R_{f,m}$, and set ε.

Initialization:

Model the railway infrastructure network via $S_r, S'_r, C_m, N^s, N^m, N_r, MOT_m$;

Model the train traffic flows and maintenance tasks via $B_r, B^a, B^+_{r,n}$, $B^-_{r,n}, B_c, B_i, C_b, L_b, N^b, N_r^{ta}$;

Solve **TSRWM**, get $B_{r,initial}$ and obj_{TSRWM};

Set $obj_{\text{lower}} = obj_{\text{TSRWM}}$;

Solve **TSTMTS** with $B_{r,initial}$, get $\widetilde{t_m^{start}}, \widetilde{t_m^{end}}, \tilde{y}_{r,b}^{entry}, \tilde{y}_{r,b}^{exit}$, and obj_{TSTMTS};

Set $obj_{upper} = obj_{\text{TSTMTS}}$.

Iterative process:

While $\left(\left({obj_{\text{upper}} - obj_{\text{lower}}}\big/{obj_{\text{lower}}} > \varepsilon\right) \&\& \left(t_{max} > t_c\right)\right)$ **do**

Use $\widetilde{t_m^{start}}, \widetilde{t_m^{end}}$ obtained from TSTMTS and generate Constraints (57) and (58);

Use $\tilde{y}_{r,b}^{entry}, \tilde{y}_{r,b}^{exit}$ obtained from TSTMTS and generate $R_{f,m}$;

If Strategy 1

Constraints (59) are generated;

End if

If Strategy 2

$t_{phase,max} = \min\{t_{TSR,max}, t_{max} - t_c\}$;

Else

$t_{phase,max} = t_{max} - t_c$;

End if

While ($t_{phase} < t_{phase,max}$) **do**

Solve **TSR** with Constraints (57)-(59), get obj_{TSR} and $B_{r,re-route}$;

If $obj_{\text{TSR}} < obj_{\text{upper}}$

Set $obj_{\text{upper}} = obj_{\text{TSR}}$;

End if

End while

$t_{phase,max} = t_{max} - t_c$.

While $(t_{phase} < t_{phase,max})$ **do**

Solve **TSTMTS** with $B_{r,re-route}$, get $\widetilde{t_m^{start}}, \widetilde{t_m^{end}}, \tilde{y}_{r,b}^{entry}, \tilde{y}_{r,b}^{exit}$, and obj_{TSTMTS} ;

If $obj_{\text{TSTMTS}} < obj_{\text{upper}}$

Set $obj_{\text{upper}} = obj_{\text{TSTMTS}}$;

End if

End while

Itr ← *itr* +1.

End while

Output: Best solution found, including $y_{r,b}^{entry}$, $y_{r,b}^{exit}, x_{r,b}, t_{r,s}^{stop}, t_m^{start}, t_m^{end}$, obj_{upper}, and t_c

Fig. 9. Pseudo-code for the iterative algorithm

6 Computational experiments

This section presents the results obtained for a set of experiments on a realistic case study based on the railway data introduced in 2016 PSC. The experiments on the models and algorithms proposed in this paper have been performed on a desktop computer with i7-7700 @ 3.6 GHz CPU and 16.0 GB RAM. The MILP model proposed for the ITTTMTSP, introduced in Section 4, and the MILP model of each phase of the iterative algorithm, introduced in Section 5, are solved with Gurobi 7.5 on Windows 10.

In the following, Section 6.1 introduces the test case with a description of the railway infrastructure, maintenance tasks and traffic flows. Section 6.2 gives a validation of the ITTTMTSP model with detailed information on the characteristics of the solution provided by the resolution of the model. Section 6.3 quantifies the potential benefits of using the integrated optimization model versus a sequential (non-integrated) solution method. Section 6.4 presents the results obtained by solving the integrated optimization model and by various versions of the iterative algorithm. Section 6.5 gives a quantitative comparison between the performance of the first place of 2016 PSC and the best version of our iterative algorithm in terms of solution quality (best solutions by the approaches) and computation time.

6.1 Description of the test case

The railway network is shown in Fig. 10. This network has been presented during the 2016 PSC and consists of 27 stations, 55 segments, 261 block sections, 1027 cells, 1811 links and 1619 nodes. The network can be divided in five parts: western, eastern, northern, southern, and M station. The latter part is the most complex station area of this network, including 19 siding tracks and 4 main tracks, is connected to the other four parts. For this reason, most of the turn-around block sections are in M station. For detailed information on nodes, links, cells, block sections, maintenance tasks and trains, we refer the reader to the dataset published in the 2016 PSC.

Table 5 presents maintenance and traffic flow data regarding the three cases investigated in the 2016 PSC for the network of Fig. 10. The maintenance cells are located either in stations or track segments. Specifically, Fig. 10 reports the cells in which maintenance tasks can be performed for the three cases investigated in this paper. The maintenance tasks in stations lead to train speed restriction requirements, while the maintenance tasks in the track segments result in train speed reduction requirements. By comparing the number of trains and maintenance cells, the three cases present the same number of trains while the number of maintenance cells increases significantly from case 1 to case 2 and from case 2 to case 3. The differences between the three cases will be investigated in the next subsection to assess the impact of maintenance cells on the computational complexity of solving the resulting ITTTMTSP.

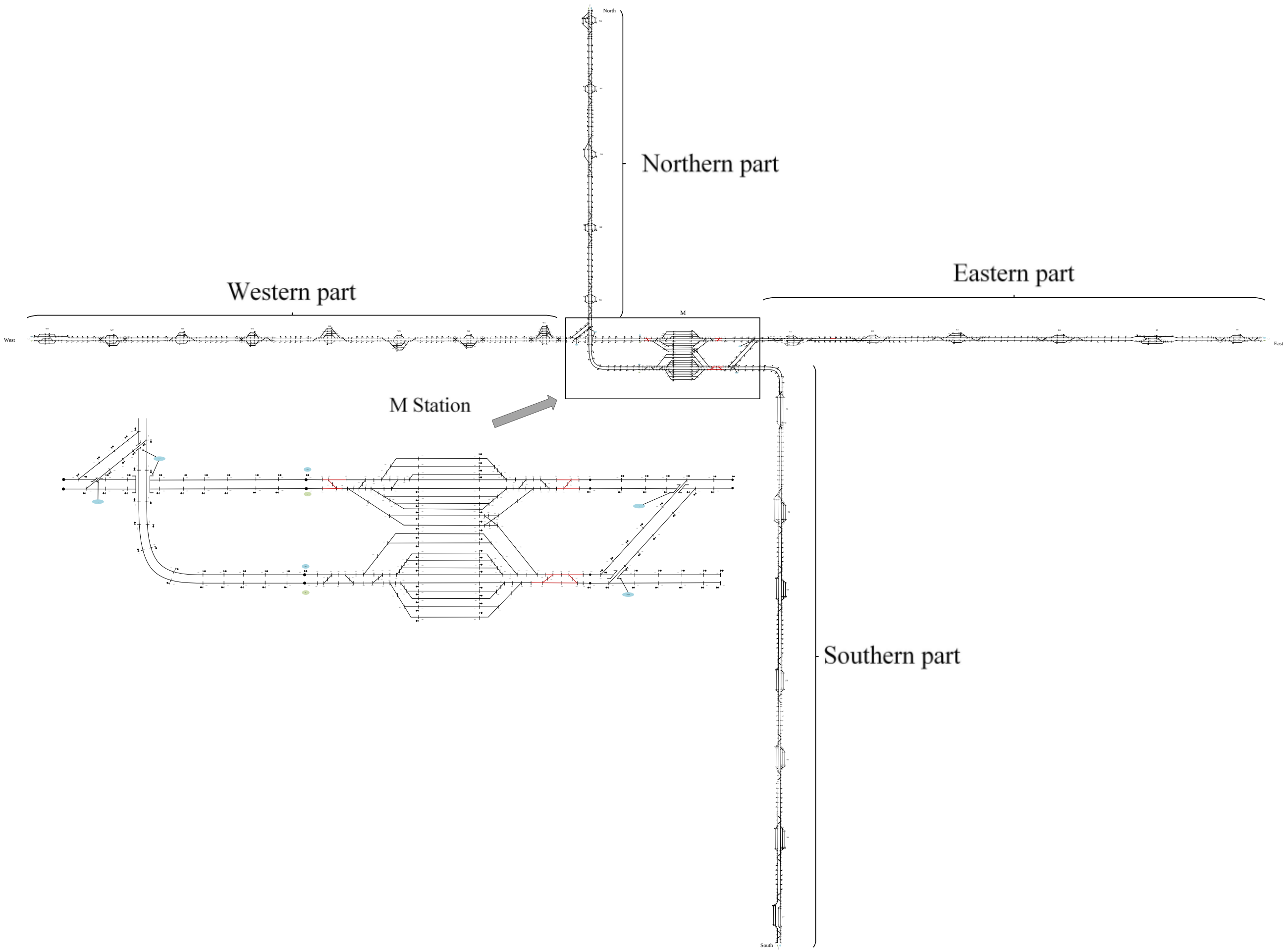


Fig. 10. Detailed layout of the studied railway network (INFORMS RAS 2016 PSC)

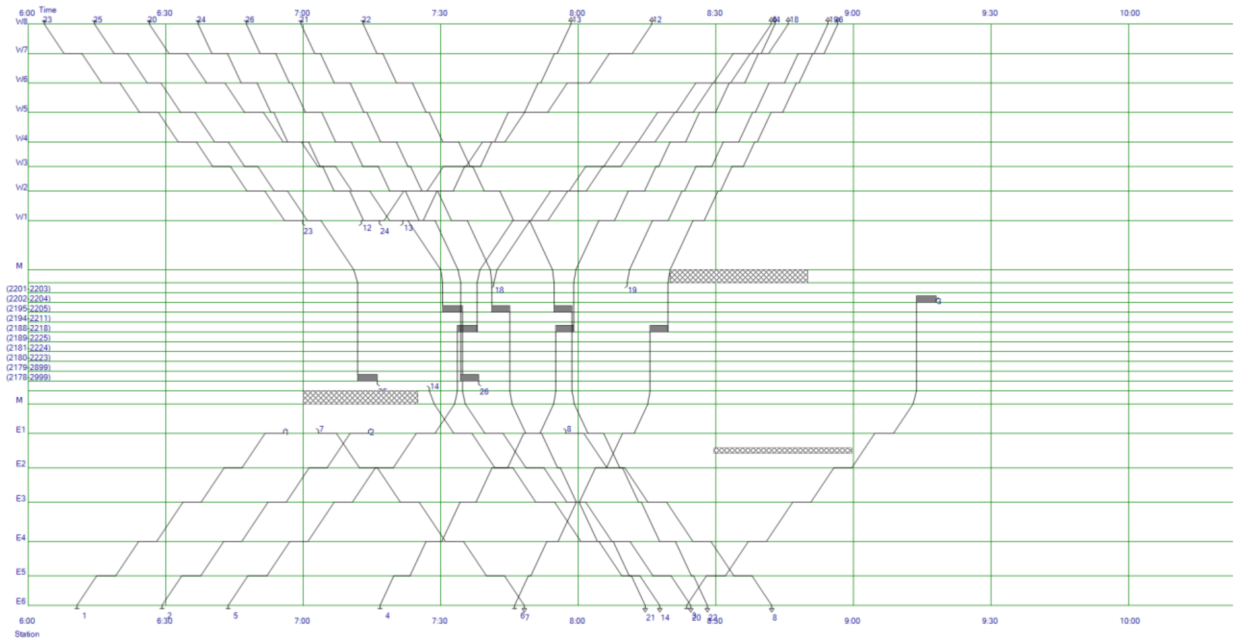
Table 5 Maintenance and traffic flow data

	Number of maintenance cells		Number of trains
	In stations	In track segments	
Case 1	2	0	26
Case 2	4	0	26
Case 3	8	1	26

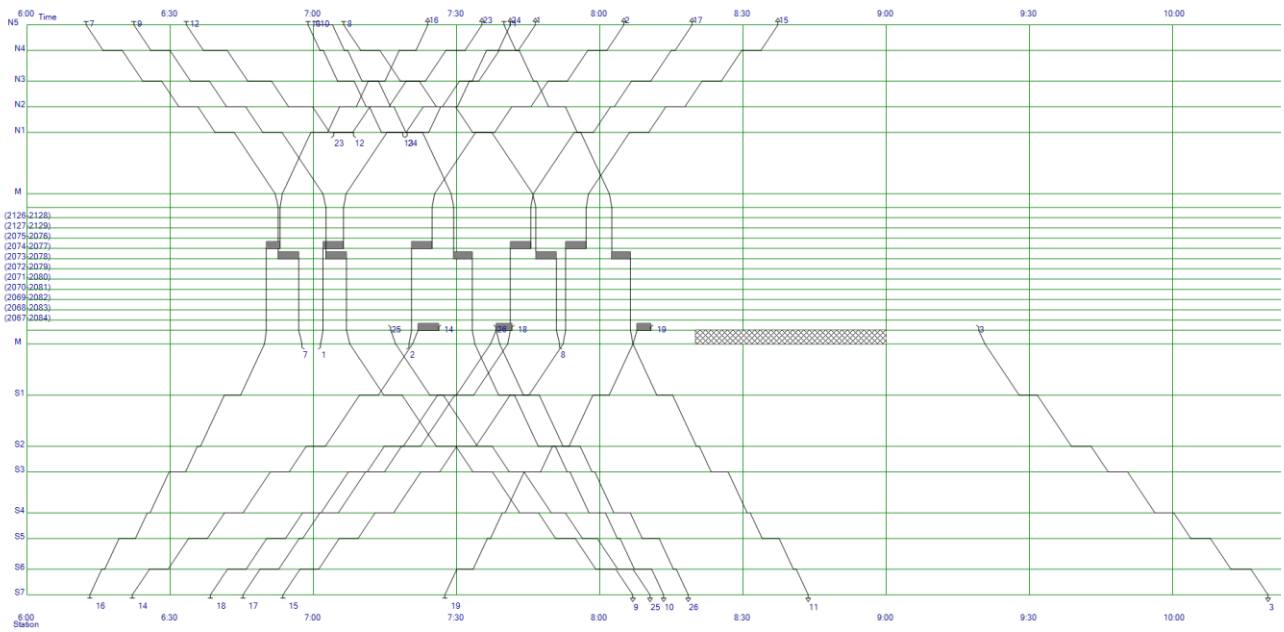
Since we aim to compare our best ITTTMTSP solutions with those provided by the first-place team in the 2016 PSC, we set the parameter values to be the same as those used in the competition. First, we simplify the blocking times by considering the running time only, i.e. we set the reservation time $t_{r,b}^{res}$ and the release time $t_{r,b}^{rel}$ for each pair train and block section equal to 0. Second, we use the maximum allowed speed profile to calculate the minimum running time of each trains in each link, i.e. the recovery time $t_{r,l}^{reco}$ regarding each train in each link is set equal to 0. Third, we do not consider the influence of the train length on the computation of blocking times, since each train is considered as a virtual dot in the 2016 PSC.

6.2 Validation of the ITTTMTSP model

We analyze a case 3 solution computed by Gurobi 7.5 for the mathematical model presented in Section 4 to validate the correctness of the proposed ITTTMTSP solution. The corresponding space-time diagram of the train timetable and the schedule of maintenance tasks are illustrated in Fig. 11(*a*) for the 15 stations (W1, …, W8, M, E1, …, E6) in the western and eastern parts of the rail network of Fig. 10 and in Fig. 11(*b*) on the 13 stations (N5, …, N1, M, S1, …, S7) in the northern and southern parts of the rail network of Fig. 10.



(*a*) Timetable of trains and maintenance tasks on the 15 stations in the western and eastern parts of the rail network



(*b*) Timetable of trains and maintenance tasks on the 13 stations in the northern and southern parts of the rail network

Fig. 11 A feasible ITTTMTSP solution obtained by our model for case 3

We observe that all the constraints of the ITTTMTSP are satisfied in the solution reported in Fig. 11. Specifically, a feasible route is selected for each train in the rail network, the potential conflicts between trains are solved, the dwell time, running time and turn-around constraints are satisfied. The maintenance tasks (see the shaded areas in Fig. 11) are scheduled within their required time window and their duration time is equal to or larger than their minimum required duration time. The space-time trajectories of all trains satisfy the constraints related to the scheduled maintenance tasks.

From the solution representation of Fig.11, it is hard to observe the satisfaction of train speed reduction and restriction constraints due to the scheduled maintenance tasks. For this reason, we next give a more detailed description of the network layout nearby the maintenance cells and comment on some required train speed profile adjustments.

Fig.12 shows the microscopic representation of the railway infrastructure nearby maintenance tasks scheduled in case 3. This infrastructure is part of M station of Fig. 10. The maintenance links are highlighted in red color with dotted lines. Specifically, the maintenance task contains two maintenance cells (cells 266 and 291) and six maintenance links (three links 2241-2242, 2243-2242 and 2242-2246 in cell 266 plus three links 2247-2244, 2243-2244 and 2244-2245 in cell 291). This maintenance task lasts 25 minutes, starts at 7:00:00 and ends at 7:25:00.

Regarding the train speed profile adjustments required in the solution of Fig. 11, we consider e.g. train 14 moving in the network from node 3000 to node 2248. This is the first train traveling through maintenance cells 266 and 291. According to speed restriction constraints (39), the running time of train 14 on block section 311 increases from 57 seconds (minimum running time) to 63 seconds (feasible running time). With these constraints, the restrained speed of the first train on the maintenance links is equal to the minimum value between the link speed limit multiplied by the train speed multiplier and the speed limit of the first train. Specifically, $\min\{120 \times 0.7, 40\} = 40$ mph gives the restrained speed of train 14 on maintenance links 2245-2244 and 2242-2246, while $\min\{30 \times 0.7, 40\} = 21$ mph gives the restrained speed of train 14 on maintenance links 2244-2243 and 2243-2242. On maintenance links 2247-2244 and 2244-2245, train 5 is the second train traveling through maintenance cell 291 with a route going from node 2249 to node 2226. This train must satisfy speed restriction constraints (40). However, the restrained speed of train 5 on these maintenance links is almost the same as its maximum allowed speed. Hence, the running time of train 5 on these maintenance links does not change in the solution of Fig. 11. Since the train speed profiles are not affected by speed reduction constraints in this solution for case 3, their effect is not discussed here.

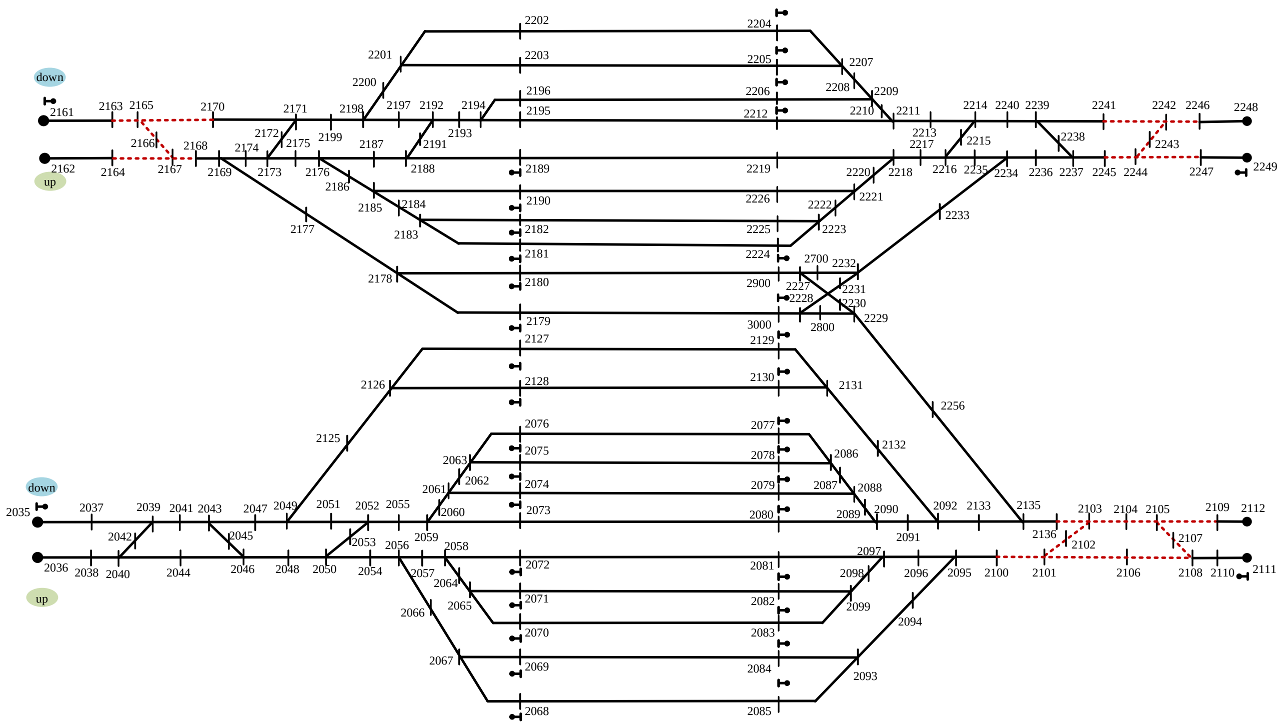


Fig. 12 Infrastructure nearby maintenance tasks scheduled in case 3

6.3 Benefits of integrating TT and TMTS

In Fig. 13, an integrated optimization method based on the ITTTMTSP model **(ITTTMTSPM)** developed in Section 4 is compared with the sequential and random (both non-integrated) methods. The latter two methods are used to compute TT and TMTS solutions by means of the following two-step approach: 1) for the sequential method, we assign the earliest possible start and end times to each maintenance task; for the random method, we generate the start times in a randomly distributed manner within a time window $[mot_m^s, mot_m^e]$ for each track maintenance task m, while the corresponding end times are calculated by adding the minimum duration times to the start times; 2) we then solve the TT by using the MILP model of the TSR phase of the algorithmic framework described in Section 5. Both ITTTMTSPM and the MILP models in the TSR phase are solved by Gurobi 7.5 with a time limit of computation of 5 hours.

The comparison of Fig. 13 is based on the resolution of the three cases introduced in Section 6.1. For the random method, we performed five experiments for each case by varying the start times for each track maintenance task as described above. The best and worst results obtained on the random method are shown in in Fig. 13. For each case, we report the value of the objective function of the ITTTMTSP (in seconds). The results highlight the positive effect of combining TT and TMTS: ITTTMTSPM outperforms the sequential and random methods. The improvement is more evident for case 3, since this is the case with the largest number of maintenance cells (as reported in Table 5). We observe that even though the best results of the random method are close to that of ITTTMTSPM, the worst results of the random method turn out to be much larger to the other two methods, especially for cases 2 and 3.

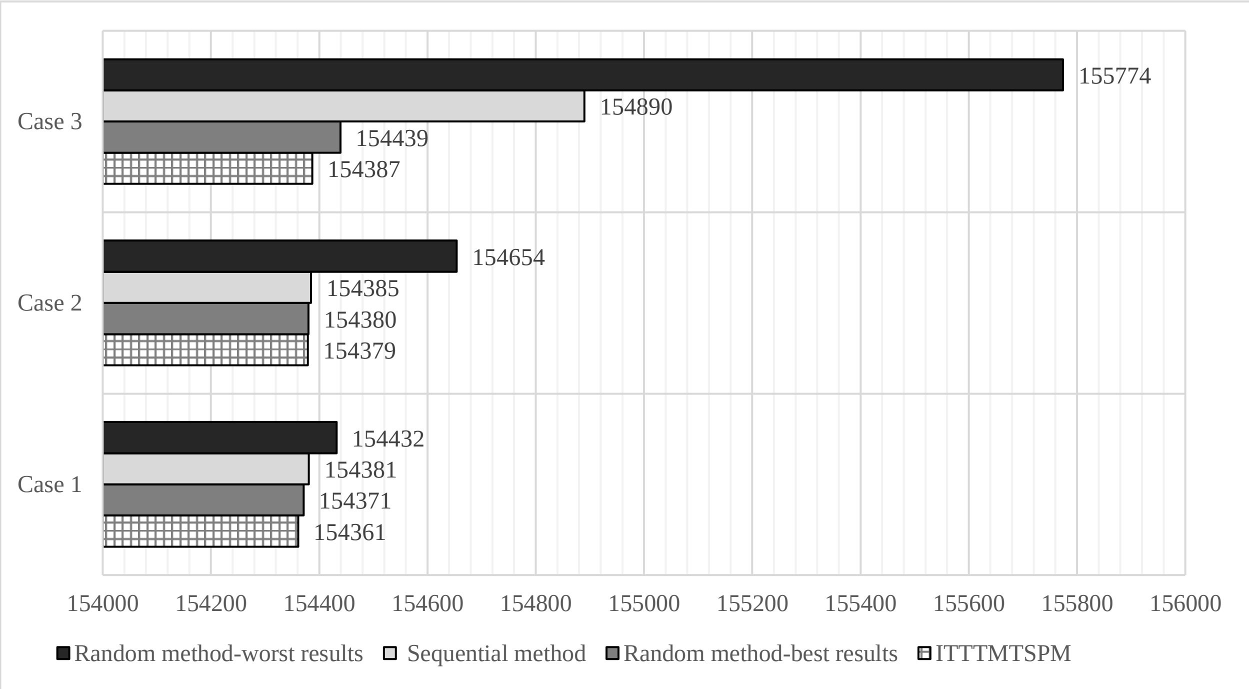


Fig. 13 Comparison (objective function, in seconds) of ITTTMTSPM versus sequential and random methods

6.4 Efficiency and effectiveness of the iterative algorithm

This sub-section focuses on investigating the quality of the solutions provided by the iterative algorithm and the computation time required by this algorithm. We present the results obtained on the three cases of Section 6.1 with 26 trains. For each case, we also consider two simplified situations with 10 and 20 trains to compare the solutions provided by the iterative algorithm with the ones computed by Gurobi 7.5 for the ITTTMTSPM with a 5-hour time limit of computation. Specifically, we present the performance of the following four versions of the iterative algorithm: **IA** is the algorithm of Fig. 10 without speed-up strategies; **IA-1** is IA with Strategy 1 and $t_m = 3600$s; **IA-2** is IA with Strategy

2 and $t_{TSR,max}$ = 300s; **IA-1-2** is IA with Strategy 1 and t_m = 3600s plus Strategy 2 and $t_{TSR,max}$ = 300s. For all versions of the iterative algorithm, we use the following parameters: t_{max} is 3600s, ε is 0.05. For IA and IA-1, the TSR phase is solved with no consideration of $t_{TSR,max}$. All the phases of the iterative algorithm are solved by Gurobi 7.5.

Table 6 reports the quantitative comparison between ITTTMTSPM, IA, IA-1, IA-2, and IA1-2. For each investigated case, we report the objective function value (*Obj Value*, in seconds), the total computation time (*CPU Time*, in seconds), and the optimality gap (*Opt Gap*, in percentage). The bold numbers in each row regarding Obj Value highlight the best-known solutions computed by the corresponding solution method. For each case (i.e. for each column), the underlined number regarding CPU Time shows the total computation time of the fastest solution method. Opt Gap indicates the percentage difference between the best solution found by each solution method and the best-known lower bound, i.e. the best lower bound between the one achieved for ITTTMTSPM and the one computed by our iterative algorithm.

Table 6 Comparison of the results obtained by different solution methods

Studied Case		Case 1			Case 2			Case 3		
Number of Trains		10	20	26	10	20	26	10	20	26
ITTT	Obj Value (sec)	**60754**	**121346**	**154361**	**60760**	121354	154379	**60772**	121366	154387
MTSP	CPU Time (sec)	3.47	40.01	1210.37	12.76	18000	18000	27.49	18000	18000
M	Opt Gap (%)	0.0000	0.0000	0.0000	0.0000	0.0066	0.0117	0.0000	0.0165	0.0168
	Obj Value (sec)	**60754**	**121346**	154373	**60760**	**121352**	154381	**60772**	**121364**	**154379**
IA	CPU Time (sec)	12.84	90.5	3600	16.55	190.77	3600	32.69	3600	3600
	Opt Gap (%)	0.0000	0.0000	0.0078	0.0000	0.0049	0.0130	0.0000	0.0148	0.0117
	Obj Value (sec)	**60754**	**121346**	154363	**60760**	**121352**	**154367**	**60772**	121368	154381
IA-1-2	CPU Time (sec)	12.09	92.4	665.08	13.25	100.59	477.12	21.02	643.04	1012.37
	Opt Gap (%)	0.0000	0.0000	0.0013	0.0000	0.0049	0.0039	0.0000	0.0181	0.0130
	Obj Value (sec)	**60754**	**121346**	**154361**	**60760**	**121352**	**154367**	**60772**	**121364**	154389
IA-1	CPU Time (sec)	12.03	91.53	1363.08	13.19	100.15	3600	21.36	3600	3600
	Opt Gap (%)	0.0000	0.0000	0.0000	0.0000	0.0049	0.0039	0.0000	0.0148	0.0181
	Obj Value (sec)	**60754**	**121346**	**154361**	**60760**	**121352**	**154367**	**60772**	**121364**	**154379**
IA-2	CPU Time (sec)	12.42	87.31	526.06	15.86	182.6	516.08	26.11	548.06	664.89
	Opt Gap (%)	0.0000	0.0000	0.0000	0.0000	0.0049	0.0039	0.0000	0.0148	0.0117

From the results in Table 6, ITTTMTSPM is the solution method that finds the smallest number (5) of best-known solutions (bold values of Obj Value) compared with the four versions of the iterative algorithm, even if we set the largest time limit of computation (18000 seconds) for the latter method. This comparison of ITTTMTSPM with the four versions of the iterative algorithm highlights the high added value of the algorithmic framework proposed in Section 5. However, ITTTMTSPM is a competitive method for all the cases with less (10) trains or less (2) maintenance cells, both in terms of solution quality and computation time. We conclude that when the number of trains and/or maintenance cells increases there is a clear need to use customized algorithms to solve the ITTTMTSP in a short computation time.

When comparing the various versions of the iterative algorithm, IA (i.e. the version without speed-up strategies) presents the worst quality solutions for cases 1 and 2 with 26 trains and takes the longest computation time (1 hour) for case 3 with 20 and 26 trains. This comparison confirms the relevance of developing speed-up strategies in this paper. When adding Strategy 1, IA-1 improves (worsens) the performance of the iterative algorithm compared to IA for two cases (one case) in terms of solution quality. Similarly, IA-1-2 improves (worsens) the performance of the iterative algorithm compared to IA for two cases (two cases) in terms of solution quality. This fluctuating trend is due to two concurrent factors related to Strategy 1, i.e. to the reduction of the number of potential first and second trains in each maintenance cell. A positive factor is a slight reduction of the number of maintenance-related constraints, while a negative factor is a limited impact in terms of smartly reducing the search space. We conclude that Strategy 1 does not sufficiently help to improve the performance of the iterative algorithm. On the other hand, Strategy 2 is very promising, since the computation time is better distributed among the TSTMTS and TSR phases of the iterative algorithm. With the latter strategy, IA-2 outperforms all the other methods in terms of solution quality and is often the fastest algorithm on the most complex instances of Table 6. The optimality gap related to the best-known solutions computed by IA-2 is very low. In the next subsection we will compared the performance of IA-2 with the best solutions provided during the 2016 PSC.

6.5 Comparison with the solutions of the first-place team in 2016 PSC

Table 7 compares the performance of our algorithm IA-2 with the constraint programming method proposed by the team that got the first prize in 2016 PSC (Carpov et al. 2016). Specifically, the computational results reported in the latter approach by Carpov et al. (2016) were obtained by on a single core of AMD Opteron 6172 processor (2.1GHz). The comparison is shown in terms of the three indicators used in Table 6, i.e. Obj Value, CPU Time and Opt Gap. Regarding the computation of the latter indicator, we use the best-known lower bound obj_{lower} and the left part of expression (60) both for our iterative algorithm and the solution method developed in Carpov et al. (2016). We report the comparison on the three cases presented during the 2016 PSC, i.e. cases 1, 2 and 3 of Table 5 with 26 trains.

Table 7 Comparison of IA-2 with the best solutions provided by the first-place team in 2016 PSC

	Carpov et al. 2016			**IA-2**		
Instance	Obj Value (sec)	CPU Time (sec)	Opt Gap (%)	Obj Value (sec)	CPU Time (sec)	Opt Gap (%)
Case 1	155294	7200	0.6044	154361	526.06	0.0000
Case 2	156865	7200	1.6222	154367	516.08	0.0039
Case 3	158985	7200	2.9956	154379	664.89	0.0117

The results of Table 7 clearly show that IA-2 outperforms the method proposed by Carpov et al. (2016), both in terms of solution quality and computation time. We remark that, even if we used a slightly better desktop computer to perform the experiments, the difference in terms of computation time between the two approaches is highly significant, i.e. from 2 hours to around 11 minutes. Furthermore, the solution quality is also significantly better for IA-2, i.e. 933s, 2498s and 4606s for cases 1, 2 and 3 respectively. The new best-known solutions computed by IA-2 are provided in Appendix.

6.6 Scalability of the proposed methods

We now investigate the ability of Gurobi 7.5 and of the best version of our iterative algorithm to solve the ITTTMTSP, i.e. ITTTMTSPM and IA-2 respectively. The maximum time limit of computation is set to 5 hours for ITTTMTSPM and to 1 hour for IA-2. We consider larger instances compared to 2016 PSC. These are obtained by considering case 3 and by varying the number of scheduled trains between 10 and 38. Table 8 reports the comparison between ITTTMTSPM and IA-2, in terms Obj Value (in seconds), CPU Time (in seconds) and Opt Gap (in percentage). The bold values are the best-known solutions, while the optimality gap is computed by using the best-known lower bound.

Table 8 Comparison of the results obtained by IA-2 and ITTTMTSPM for case 3

Number of Trains		10	20	26	30	34	38
ITTTMTSPM	Obj Value (sec)	**60772**	121366	154387	173711	-	-
	CPU Time (sec)	27.49	18000	18000	18000	-	-
	Opt Gap (%)	0.0000	0.0165	0.0168	0.2337	-	-
IA-2	Obj Value (sec)	**60772**	**121364**	**154379**	**173390**	**193027**	**213266**
	CPU Time (sec)	26.11	548.06	664.89	3600	3600	3600
	Opt Gap (%)	0.0000	0.0148	0.0117	0.0485	0.0581	0.3491

The results of Table 8 show that IA-2 outperforms ITTTMTSPM both in terms of solution quality and time to deliver the best-known solution. Specifically, the optimality gap is IA-2 is always small, even for the instances with 38 trains. Furthermore, ITTTMTSPM computes a solution for instances up to 30 trains with the given time limit of computation.

6.7 Influence of additional track maintenance tasks

The scalability of the algorithm is further investigated in this section on a new set of instances. We test the performance of our methods by increasing the number of track maintenance tasks as well as by introducing more tighter speed profiles with respect to the most complicated case of the 2016 PSC, i.e. Case 3.

Table 9 shows key information on the three additional track maintenance tasks introduced in Case 3 of the 2016 PSC, named additional tasks 1, 2 and 3. These tasks involve four cells of the railway network in Fig. 10. Additional tasks 1 and 2 are in the southern and western segment parts of the network respectively, while additional task 3 is in the bottom-left part of M station. In addition, the values of v_m for additional tasks 1 and 2 are reduced from 80 mph (as specified by the 2016 PSC) to 40 mph. Table 9 also specifies the involved cells, the time windows (in seconds) of the track maintenance task start times, and the minimum duration times (in seconds) for the three additional track maintenance tasks.

Table 9 Key information of the three additional track maintenance tasks

Track maintenance task	Location	Cells	Start time window (sec)	Minimum duration time (sec)	v_m (mph)
Additional task 1	Southern segment part	{363}	[4800, 16800]	1000	40
Additional task 2	Western segment part	{792}	[4800, 16800]	1000	40
Additional task 3	M station	{161, 222}	[3600, 15600]	1000	-

Table 10 gives the results obtained on the new set of 24 instances generated from Case 3 as follows. We introduce four cases (Cases 4-7) of increasing complexity and for each case we consider 10, 20, 26, 30, 34 and 38 trains. Specifically, Case 4 includes the additional track maintenance task 1, Case 5 contains the additional track maintenance task 2, Case 6 includes both the additional track maintenance tasks 1 and 2, and Case 7 consists of the additional track maintenance tasks 1, 2 and 3. Each case is solved via ITTTMTSPM and the iterative algorithm IA-2. For consistency with the other experiments, the maximum time limit of computation is set to 1 hour for IA-2 and to 5 hours for ITTTMTSPM.

We observe from Table 10 that ITTTMTSPM cannot obtain a feasible solution for all the test cases in which the number of trains is larger than or equal to 26. By contrast, the iterative algorithm IA-2 computes near-optimal solutions for all the test instances, with a maximum optimality gap equal to 1.1274%. Furthermore, IA-2 also outperforms ITTTMTSPM for all the test cases with 20 trains. Overall, the computational results in Table 10 demonstrate the reliability and effectiveness of our iterative algorithm.

Table 10 Comparison of the results obtained by IA-2 and ITTTMTSPM with additional track maintenance tasks

		Number of Trains	10	20	26	30	34	38
Case 4 (Case 3 + additional task 1)	**ITTTMTSPM**	Obj Value (sec)	**60772**	121368	-	-	-	-
		CPU Time (sec)	122.96	18000	-	-	-	-
		Opt Gap (%)	0.0296	0.0181	-	-	-	-
	IA-2	Obj Value (sec)	**60772**	**121364**	**154379**	**173402**	**193104**	**213694**
		CPU Time (sec)	47.21	508.46	694.22	3600	3600	3600
		Opt Gap (%)	0.0296	0.0148	0.0117	0.0554	0.0980	0.5505
Case 5 (Case 3 + additional task 2)	**ITTTMTSPM**	Obj Value (sec)	**60772**	121366	-	-	-	-
		CPU Time (sec)	153.88	18000	-	-	-	-
		Opt Gap (%)	0.0296	0.0165	-	-	-	-
	IA-2	Obj Value (sec)	**60772**	**121364**	**154379**	**173406**	**193113**	**213380**
		CPU Time (sec)	28.91	471.61	701.35	3600	3600	3600
		Opt Gap (%)	0.0296	0.0148	0.0117	0.0577	0.1026	0.4028
Case 6 (Case 3 + additional tasks 1 and 2)	**ITTTMTSPM**	Obj Value (sec)	**60772**	121368	-	-	-	-
		CPU Time (sec)	410.64	18000	-	-	-	-
		Opt Gap (%)	0.0296	0.0181	-	-	-	-
	IA-2	Obj Value (sec)	**60772**	**121364**	**154389**	**173503**	**193319**	**213772**
		CPU Time (sec)	38.68	559.39	704.92	3600	3600	3600
		Opt Gap (%)	0.0296	0.0148	0.0181	0.1137	0.2094	0.5872
Case 7 (Case 3 + additional tasks 1, 2 and 3)	**ITTTMTSPM**	Obj Value (sec)	**60772**	121372	-	-	-	-
		CPU Time (sec)	476.65	18000	-	-	-	-
		Opt Gap (%)	0.0296	0.0214	-	-	-	-
	IA-2	Obj Value (sec)	**60772**	**121364**	**154416**	**173651**	**193547**	**214920**
		CPU Time (sec)	55.94	479.71	698.21	3600	3600	3600
		Opt Gap (%)	0.0296	0.0148	0.0356	0.1991	0.3276	1.1274

When looking at the impact of the additional track maintenance tasks in Table 10, the optimality gap of IA-2 becomes slightly larger as the number of maintenance tasks increases, showing that the number of track maintenance tasks has a major impact on the complexity of solving the joint optimization problem.

Comparing the results of Tables 8 and 10, we also have the following observation on the problem complexity: the more tighter speed profiles adopted for the additional track maintenance tasks have led to slightly larger values of the optimality gap in Table 10, especially when the number of trains is larger than or equal to 30.

7 Conclusions and further research

This paper proposes a novel MILP formulation with tight structure to deal with the integrated optimization of train timetabling and track maintenance scheduling. In our MILP-based approach, block sections are the basic modeling units. The running time of each train, the headway time between trains and the start/end time of maintenance tasks are computed in each block section at a microscopic level. The proposed formulation captures all the 2016 PSC constraints, including train speed trajectory limitations on each link subject to maintenance tasks. From the one hand, the introduction of flexible start and end times of the maintenance tasks increases the level of detail and thus the complexity of solving the integrated optimization problem. From the other hand, the proposed approach enables a precise formulation of the speed restriction constraints for the speed trajectories of the first and second trains through the links affected by maintenance works, and the temporary speed restriction constraints for all the trains travelling in the opposite direction in the cells nearby the maintenance works. Since the resulting joint optimization problem is strongly NP-hard, we propose an iterative algorithm in order to compute near-optimal solutions for realistic railway instances. The proposed algorithm is based on a problem decomposition based on an iterative optimization of train routing, train timing and maintenance task scheduling decisions. The computational experiments on the three 2016 PSC cases show that our iterative algorithm outperforms other methods (i.e. a standard MILP solver and the results obtained by the first place in 2016 PSC) in terms of solution quality and computation time. Specifically, the iterative algorithm provides new best-known solutions for each of the 2016 PSC instances and reduces the time to deliver the best-known solutions for the most complex 2016 PSC instances significantly, from several hours to up to around 11 minutes. Furthermore, the iterative algorithm is also able to solve significantly larger instances to near-optimality.

The research on the joint optimization TT and TMTS can be extended in several interesting directions. Additional microscopic constraints (e.g. the accurate modelling of the train length) and stronger constraints (e.g. deadline constraints for the arrival time of the trains at their destination) can be modelled in the mathematical formulation of the joint optimization problem. Other settings of the experiments can be investigated to further assess the applicability of the proposed methodology. Our MILP approach minimizes the total running and dwell times of all trains without considering the preferred arrival time of the trains at their destinations. Different performance indicators can be considered that explicitly consider the arrival time window of each train to better satisfy practical timetabling requirements. Objective functions can also be studied to compute robust timetabling solutions, to optimize specific maintenance aspects, or to consider other performance indicators, such as in Samà et al. (2017b). As an important further contribution, we could incorporate the minimization of maintenance costs into the mathematical model and explore the trade-off relationship between TT and TMTS objectives. In addition, energy consumption is another key aspect to be optimized and it would be interesting to integrate energy-efficient train movements into our optimization model and methods.

Future research efforts can be dedicated to investigating other efficient problem decomposition techniques, such as time-decomposition methods. For example, one can apply a rolling horizon method combined with our iterative algorithm. Algorithmic developments can also be dedicated to the generalization of our methodology to deal with larger ITTTMTSP instances. For example, since the integrated micro-macro approach for TT in Bešinović et al. (2016) has proven to be flexible and effective, and a similar hierarchical optimization framework can be developed to manage the ITTTMTSP for a large-scale network. Within this type of framework, our mathematical model and methods can be used to test the microscopic validity of the ITTTMTSP solutions computed at macroscopic level.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 61603317), National Key Research and Development Program of China (No. 2017YFB1200700-1), and Special Fund for Basic Scientific Research of Central Universities (No. 2682017CX023). The first and third authors are deeply grateful for the financial support from the China Scholarship Council (201707000080, 201707005054). The generous support of our colleagues Gongyuan Lu and Shuguang Zhan is greatly appreciated.

References

Bešinović, N., Goverde, R.M.P., Quaglietta, E., Roberti, R., 2016. An integrated micro–macro approach to robust railway timetabling. *Transportation Research Part B: Methodological* 87, 14–32. https://doi.org/10.1016/j.trb.2016.02.004

Budai, G., Huisman, D., Dekker, R., 2006. Scheduling preventive railway maintenance activities. *Journal of the Operational Research Society* 57, 1035–1044.

Burdett, R.L., Kozan, E., 2010. A disjunctive graph model and framework for constructing new train schedules. European Journal of Operational Research 200, 85–98. https://doi.org/10.1016/j.ejor.2008.12.005

Cacchiani, V., Caprara, A., Toth, P., 2008. A column generation approach to train timetabling on a corridor. *4OR* 6, 125–142. https://doi.org/10.1007/s10288-007-0037-5

Cacchiani, V., Galli, L., Toth, P., 2014A. A tutorial on non-periodic train timetabling and platforming problems. *EURO Journal of Transportation and Logistics* 4, 285–320. https://doi.org/10.1007/s13676-014-0046-4

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., Wagenaar, J., 2014B. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological* 63, 15–37. https://doi.org/10.1016/j.trb.2014.01.009

Caimi, G., Chudak, F., Fuchsberger, M., Laumanns, M., Zenklusen, R., 2011. A New Resource-Constrained Multicommodity Flow Model for Conflict-Free Train Routing and Scheduling. *Transportation Science* 45, 212–227.

Carpov, S., Galea, F., Cudennec, L., 2016. *INFORMS RAS 2016 Problem Solving Competition (PSC) Final Report: Constraint programming based iterative heuristic for scheduling trains and maintenance tasks*. In: INFORMS Annual Meeting 2016, Railway Application Section, Problem Solving Competition, Nashville, USA.

Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and Solving the Train Timetabling Problem. *Operations Research* 50, 851–861. https://doi.org/10.1287/opre.50.5.851.362

Caprara, A., Monaci, M., Toth, P., Guida, P.L., 2006. A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics*, IV ALIO/EURO Workshop on Applied Combinatorial Optimization IV ALIO/EURO Workshop on Applied Combinatorial Optimization 154, 738–753. https://doi.org/10.1016/j.dam.2005.05.026

Community of European Railway and Infrastructure Companies (CER) (2012). Annual Report 2012-2013. *Increased efficiency through the reduction of rail track maintenance costs*. Brussels, Belgium. http://www.cer.be/sites/default/files/publication/FINAL_CER_AR2013_DEF.pdf

Cordeau, J.-F., Toth, P., Vigo, D., 1998. A Survey of Optimization Models for Train Routing and Scheduling. *Transportation Science* 32, 380.

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. https://doi.org/10.1016/j.trb.2009.05.004

Corman, F., D'Ariano, A., Marra, A.D., Pacciarelli, D., Samà M., 2017. Integrating Train Scheduling and Delay Management in Real-time Railway Traffic Control, *Transportation Research, Part E*, 105 213–239.

Corman, F. Goverde, R., D'Ariano, A. 2009. Rescheduling dense train traffic over complex station interlocking areas, *Lecture Notes in Computer Science*, In: Ahuja R.K., Möhring R.H., Zaroliagis C.D. (eds) *Robust & Online Large-Scale Optimization*, 5868, 369 – 386.

Corman F., Meng, L., 2014. A Review of Online Dynamic Models and Algorithms for Railway Traffic Control. *IEEE Transactions on ITS*, 16 (3), 1274 – 1284. doi: 10.1109/TITS.2014.2358392

Cui Y.P., Zhuang H., 2013. Analysis on Differences of Railway CTC System in China and Foreign Countries. *Railway Transport and Economy* 04, 20–25.

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(4) 405–419.

D'Ariano, A., Pacciarelli, D., Pranzo, M., 2007. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research* 183, 643–657. https://doi.org/10.1016/j.ejor.2006.10.034

D'Ariano, A., Pranzo, M., 2009. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics* 9(1) 63–84.

D'Ariano, A., Meng, L., Centulio, G., Corman, F., 2017. Integrated stochastic optimization approaches for tactical scheduling of trains and railway infrastructure maintenance. *Computers & Industrial Engineering*. https://doi.org/10.1016/j.cie.2017.12.010

Dessouky, M.M., Lu, Q., Zhao, J., Leachman, R.C., 2006. An exact solution procedure to determine the optimal dispatching times for complex rail networks. *IIE Transactions* 38 (2), 141–152.

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.

Giacco, G.L., D'Ariano, A., Pacciarelli, D., 2014. Rolling stock rostering optimization under maintenance constraints. *Journal of Intelligent Transport Systems: Technology, Planning, and Operations*, 18(1), 95–105.

Hansen, I. A., Pachl, J. (Eds.), 2014. *Railway Timetabling & Operations*. Hamburg, Germany: Eurailpress.

He, B., Liu, Y., Zhang, Y., 2016. *INFORMS RAS 2016 Problem Solving Competition (PSC) Final Report: An Integrated Optimization Method for Train Timetabling and Maintenance Scheduling Problem*. In: INFORMS Annual Meeting 2016, Railway Application Section, Problem Solving Competition, Nashville, USA.

He, B., Song, R., He, S., Xu, Y., 2014. High-Speed Rail Train Timetabling Problem: A Time-Space Network Based Method with an Improved Branch-and-Price Algorithm. *Mathematical Problems in Engineering* 2014, 1–15. https://doi.org/10.1155/2014/641562

Higgins, A., Kozan, E., Ferreira, L., 1997. Heuristic techniques for single line train scheduling. *Journal of Heuristics* 3, 43–62.

INFORMS, Railway Application Section (RAS), Problem Solving Competition (PSC), 2016. *Routing trains through a railway network: Joint optimization on train timetabling and maintenance task scheduling.* http://connect.informs.org/railway-applications/awards/problem -solving-competition/2016

Lai, Y.C., Fan, D.C., Huang, K.L., 2015. Optimizing rolling stock assignment and maintenance plan for passenger railway operations. *Computers and Industrial Engineering*, 85, 284–295.

Larsen, R., Pranzo, M., D'Ariano, A., Corman, F., Pacciarelli, D., 2014. Susceptibility of Optimal Train Schedules to Stochastic Disturbances of Process Times. *Flexible Services and Manufacturing Journal*, 26(4) 466–489.

Li, S., Dessouky, M.M., Yang, L., Gao, Z., 2017. Joint optimal train regulation and passenger flow control strategy for high-frequency metro lines. *Transportation Research Part B: Methodological* 99, 113–137.

Lidén, T., 2015. Railway Infrastructure Maintenance - A Survey of Planning Problems and Conducted Research. *Transportation Research Procedia* 10, 574–583. https://doi.org/10.1016/j.trpro.2015.09.011

Lidén, T., Kalinowski, T., Waterer, H., 2018. Resource considerations for integrated planning of railway traffic and maintenance windows. *Journal of Rail Transport Planning & Management*. https://doi.org/10.1016/j.jrtpm.2018.02.001

Liu, L., Dessouky, M.M., 2017. A decomposition-based hybrid heuristic algorithm for the joint passenger and freight train scheduling problem. *Computers & Operations Research* 87, 165–182.

Liu, S.Q., Kozan, E., 2010. Scheduling Trains with Priorities: A No-Wait Blocking Parallel-Machine Job-Shop Scheduling Model. *Transportation Science* 45, 175–198. https://doi.org/10.1287/trsc.1100.0332

Luan, X., Miao, J., Meng, L., Corman, F., Lodewijks, G., 2017. Integrated optimization on train scheduling and preventive maintenance time slots planning. *Transportation Research Part C: Emerging Technologies* 80, 329–359. https://doi.org/10.1016/j.trc.2017.04.010

Lusby, R., Larsen, J., Ryan, D., Ehrgott, M., 2011. Routing Trains Through Railway Junctions: A New Set-Packing Approach. *Transportation Science* 45, 228–245. https://doi.org/10.1287/trsc.1100.0362

Meng, L., Corman, F., Zhou X., Tang, T., 2018. Editorial Integrated optimization models and algorithms in rail planning and control. *Transportation Research, Part C: Emerging Technologies* 88, pp 87-90.

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. https://doi.org/10.1016/j.trb.2014.05.005

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

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 (5), 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(1), 58–80.

Peng, F., Kang, S., Li, X., Ouyang, Y., Somani, K., Acharya, D., 2010. A Heuristic Approach to the Railroad Track Maintenance Scheduling Problem: A heuristic approach to the railroad track maintenance scheduling problem. *Computer-Aided Civil and Infrastructure Engineering* 26, 129–145. https://doi.org/10.1111/j.1467-8667.2010.00670.x

Peng, F., Ouyang, Y., Somani, K., 2013. Optimal routing and scheduling of periodic inspections in large-scale railroad networks. *Journal of Rail Transport Planning & Management* 3(4), 163–171.

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

Samà, M., D'Ariano, A., Corman, F., Pacciarelli, D., 2017a. A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations. *Computers & Operations Research* 78 480–499. https://doi.org/10.1016/j.cor.2016.02.008

Samà, M., D'Ariano, A., D'Ariano, P., Pacciarelli, D., 2014. Optimal aircraft scheduling and routing at a terminal control area during disturbances. *Transportation Research, Part C* 47(1) 61–85.

Samà, M., D'Ariano, A., D'Ariano, P., Pacciarelli, D., 2017b. Scheduling models for optimal aircraft traffic control at busy airports: tardiness, priorities, equity and violations considerations, *Omega* 67, 81–98.

Samà, M., D'Ariano, A., Corman, F., Pacciarelli, D., 2017c. Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas, *Transportation Research, Part C* 80(1) 485–511.

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. https://doi.org/10.1016/j.trb.2016.01.005

Schlechte, T., 2012. *Railway Track Allocation: Models and Algorithms*. (Doktor) der Technischen Universitat Berlin. http://www.zib.de/schlechte/phdthesis.pdf

Shift2Rail (S2R) Joint Undertaking (2015). *Strategic Master Plan*, Brussels, Belgium. http://www.shift2rail.org/wp-content/uploads/2013/07/MAAP-final_final.pdf

Törnquist, J., Persson, J.A., 2007. N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B: Methodological* 41, 342–362. https://doi.org/10.1016/j.trb.2006.06.001

Vansteenwegen, P., Dewilde, T., Burggraeve, S., Cattrysse, D., 2016. An iterative approach for reducing the impact of infrastructure maintenance on the performance of railway systems. *European Journal of Operational Research* 252(1), 39–53.

Wang, I., Cheng, F., Fang, S., 2016. *RAS 2016 Problem Solving Competition (PSC) Final Report: Constraint programming based iterative heuristic for scheduling trains and maintenance tasks*. In: INFORMS Annual Meeting 2016, Railway Application Section, Problem Solving Competition, Nashville, USA.

Wang, Y., Tang, T., Ning, B., van den Boom, T. J. J., De Schutter, B., 2015. Passenger-demands-oriented train scheduling for an urban rail transit network. *Transportation Research Part C* 60, 1–23.

Yan, C., Yang, L., 2012. *RAS 2016 Problem Solving Competition (PSC) Final Report: Mixed-integer programming-based approaches for the movement planner problem: Model, Heuristics and Decomposition*. In: INFORMS Annual Meeting 2012, Railway Application Section, Problem Solving Competition, Phoenix, USA.

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

Yin, J., Tang, T., Yang, L., Gao, Z., & Ran, B., 2016. Energy-efficient metro train re-scheduling with uncertain time-variant passenger demands: An approximated dynamic programming approach. *Transportation Research Part B: Methodological* 91, 178–210.

Zhou, X., Zhong, M., 2007. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B: Methodological* 41, 320–341.

Appendix A. The blocking time for each type of block section

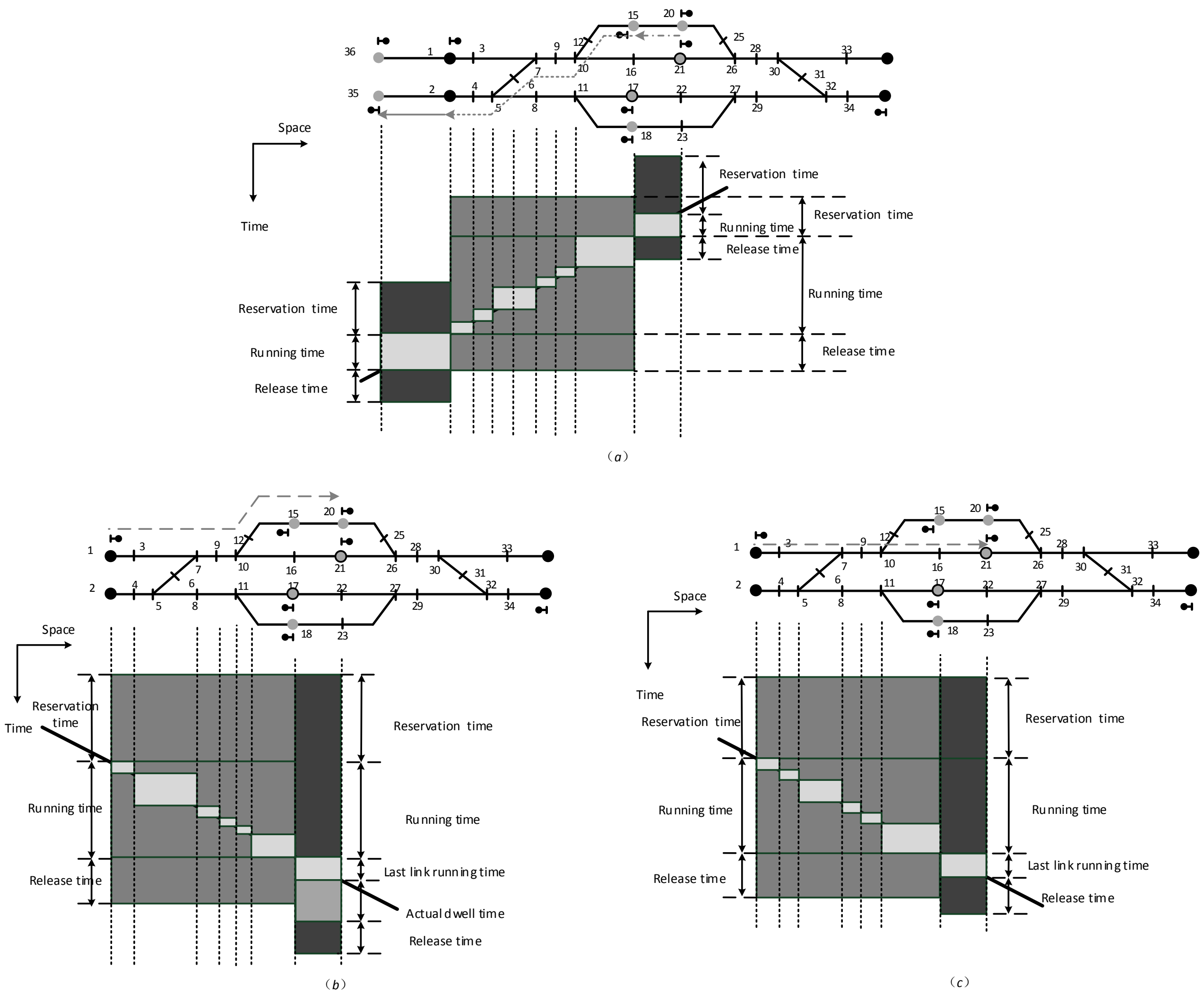


Fig. 14. Illustration on the blocking time for departure, passing and turn-around block sections (*a*), the arrival block sections ending at the siding block section node 20 (*b*) and the arrival block sections ending at the main track block section node 21 (*c*)

For the departure, passing, and turn-around block sections, the trains have no scheduled dwell time at the end of block section node, as shown in Fig 14(*a*). On the arrival block sections ending at a siding block section node, the trains stop for boarding and alighting of passengers or loading and unloading of goods. When a train arrives at the last cell (dedicated to the dwelling process) of an arrival block section, the last cell is occupied by the train, while the other cells are released, as shown in Fig. 14(*b*). The blocking time of the arrival block section does not consider the running and recovery times related to its last link. The blocking time of the latter link includes the corresponding running and recovery times plus the actual dwell time. On the arrival block sections ending at a main track block section node, trains are not allowed to stop, as illustrated in Fig. 3(*c*). Also, for this type of arrival block section, we do not directly include the running time and recovery time related to the last link in the corresponding blocking time, as shown in Fig. 14(*c*).

Appendix B. Best-known solutions

This appendix presents the ITTTMTSP solutions obtained by **IA-2** for the case studies with 26 trains described in Section 6.

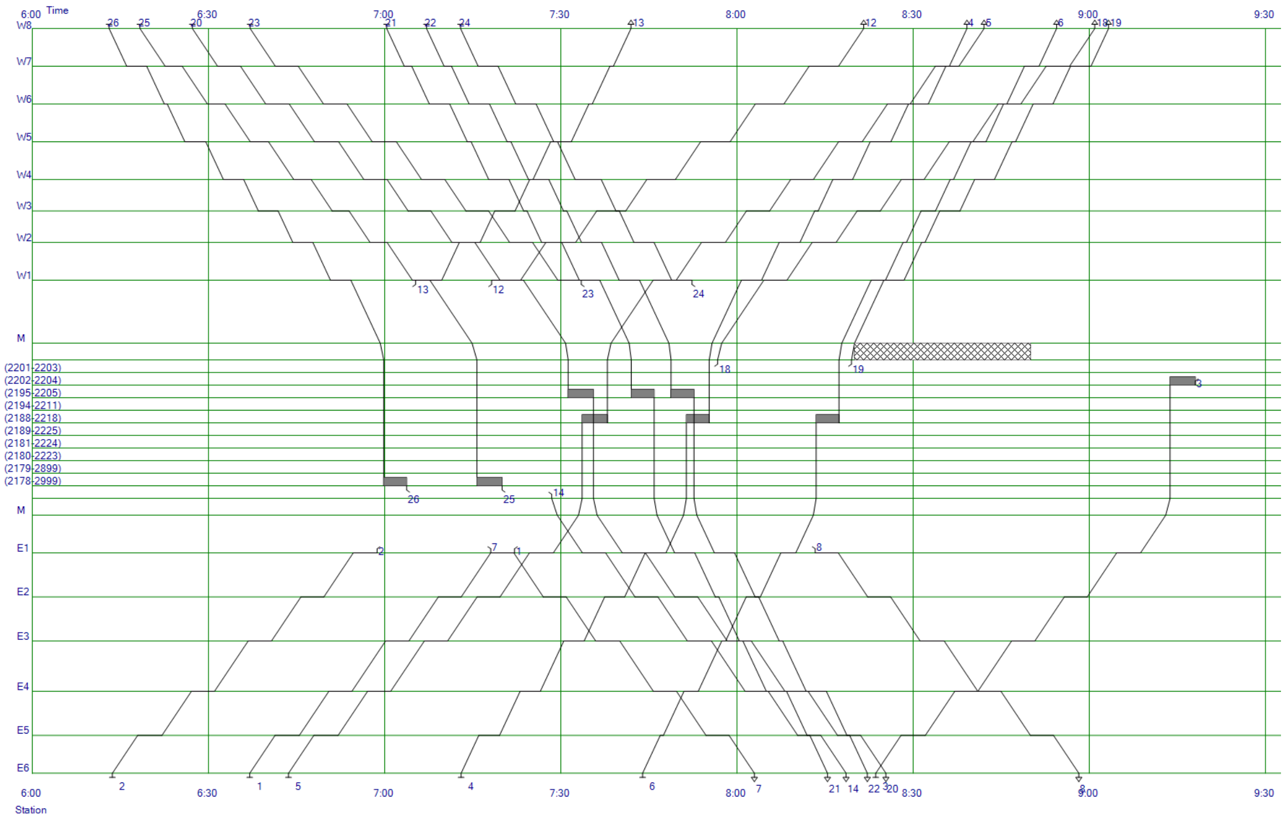


Fig.15 Timetable of trains and maintenance tasks for case 1 on the 15 stations in the western and eastern parts

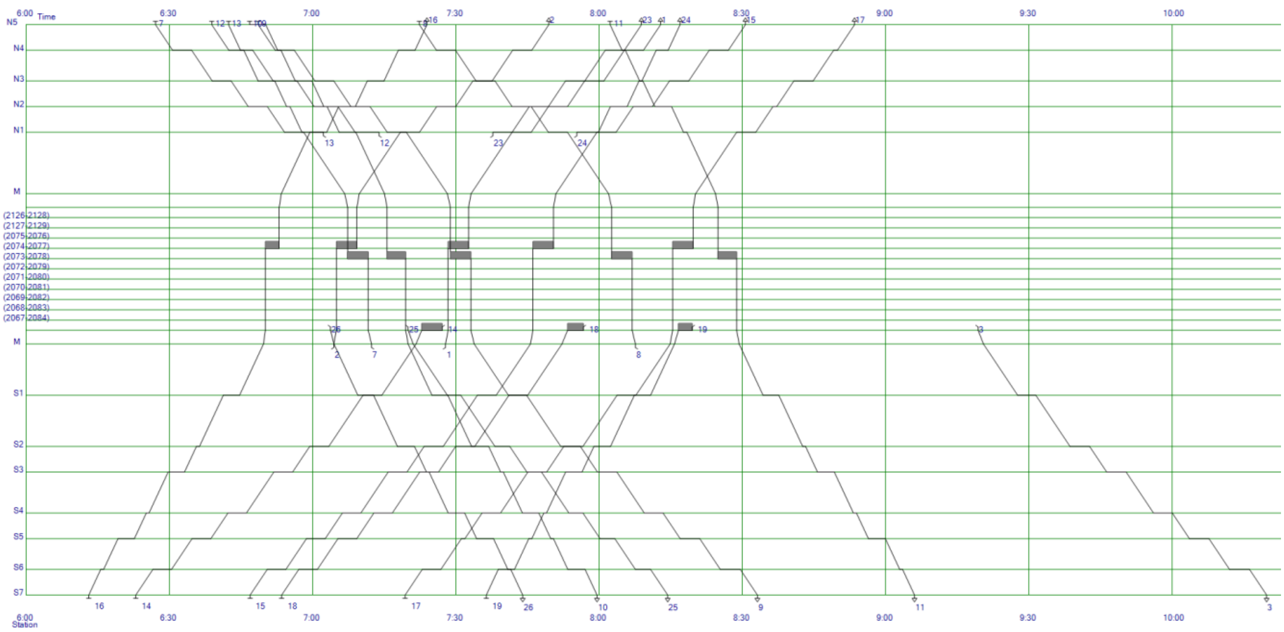


Fig.16 Timetable of trains and maintenance tasks for case 1 on the 13 stations in the northern and southern parts

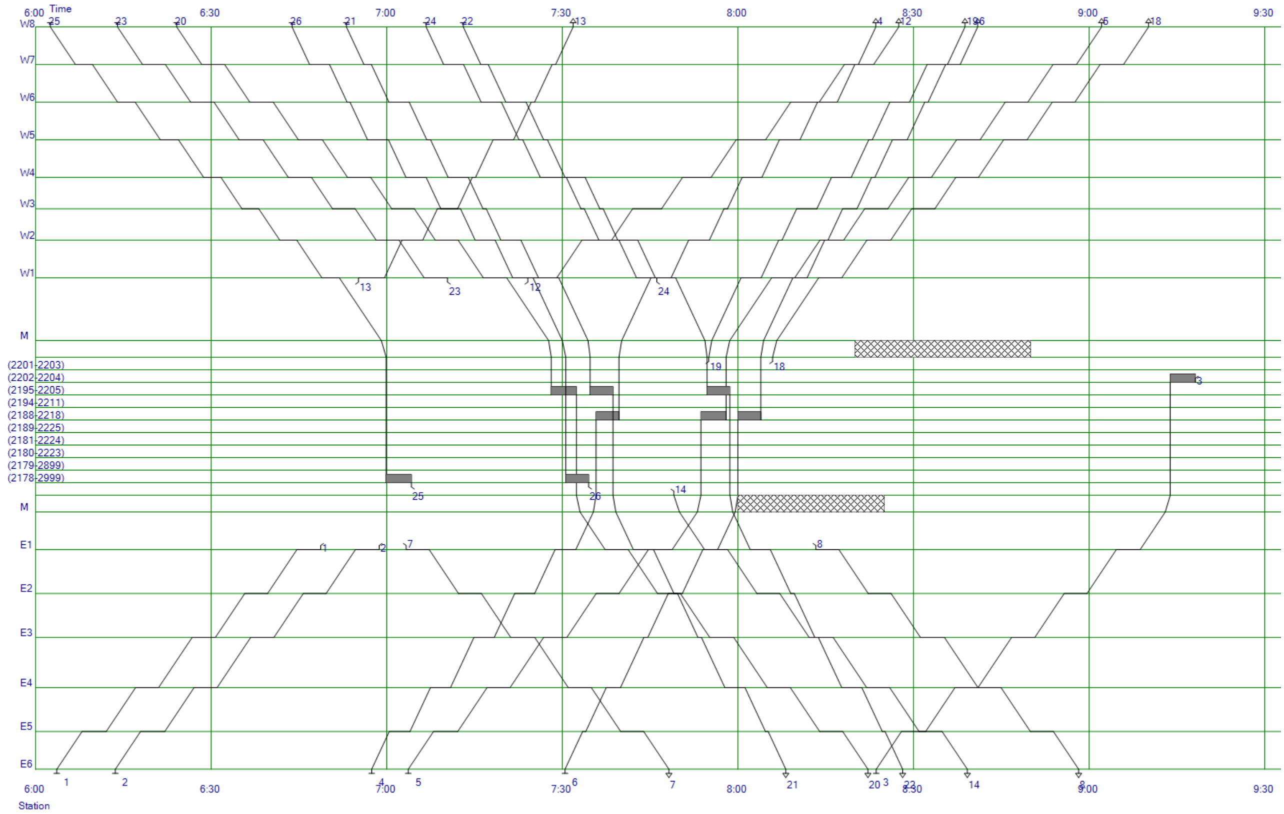


Fig.17 Timetable of trains and maintenance tasks for case 2 on the 15 stations in the western and eastern parts

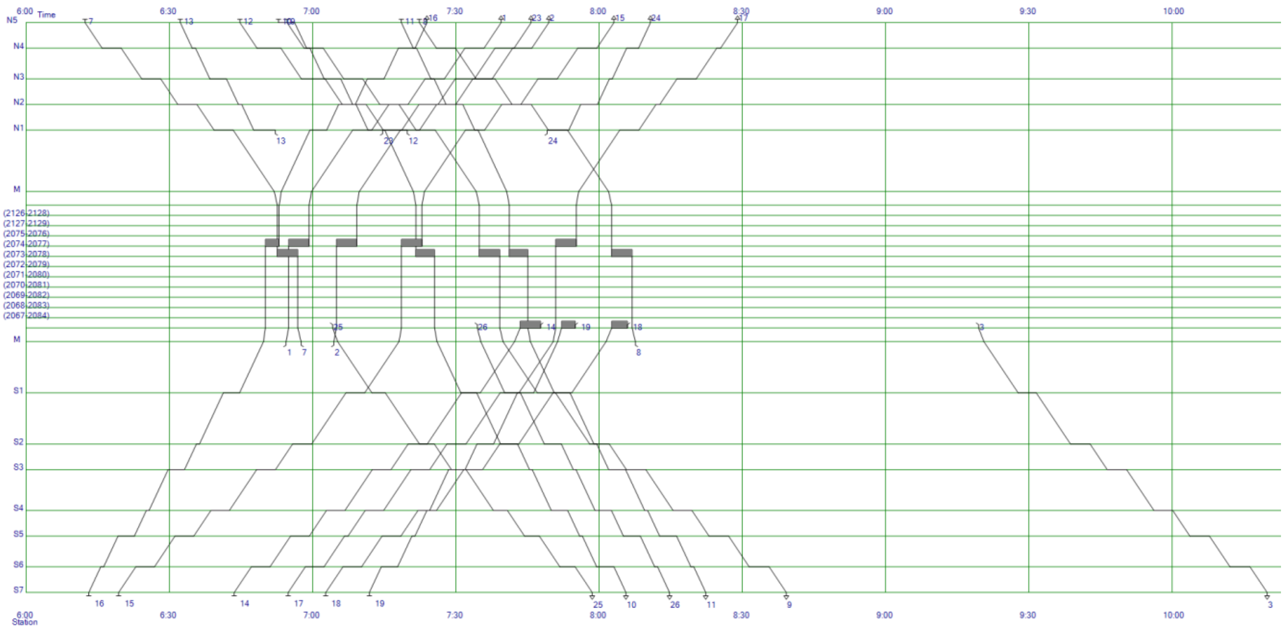


Fig.18 Timetable of trains and maintenance tasks for case 2 on the 13 stations in the northern and southern parts

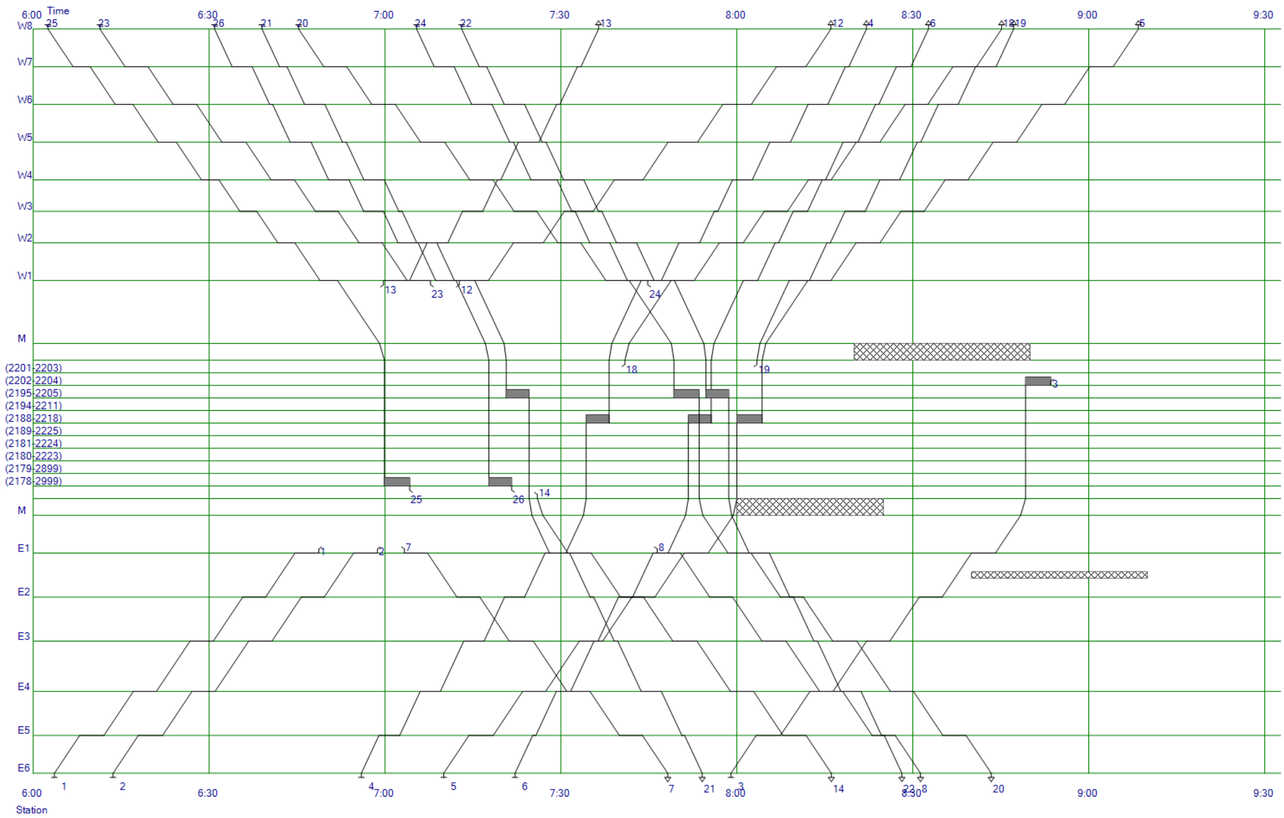


Fig.19 Timetable of trains and maintenance tasks for case 3 on the 15 stations in the western and eastern parts

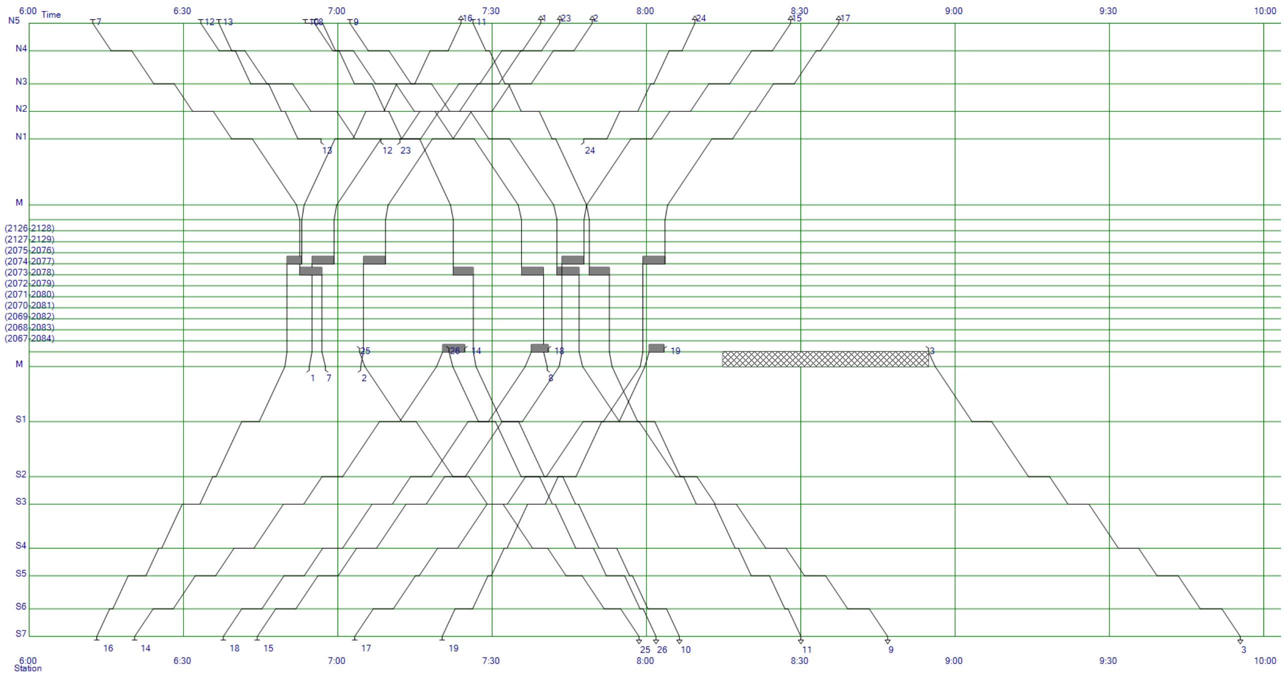


Fig.20 Timetable of trains and maintenance tasks for case 3 on the 13 stations in the northern and southern parts