Solving cyclic train timetabling problem through model reformulation: extended time-space network construct and Alternating Direction Method of Multipliers methods

Yongxiang Zhang
School of Transportation and Logistics, Southwest Jiaotong University
Chengdu 610031, China
Emails: bk20100249@my.swjtu.edu.cn, yxzhangswjtu@gmail.com

Qiyuan Peng*
School of Transportation and Logistics, Southwest Jiaotong University
Chengdu 610031, China
Email: qiyuan-peng@swjtu.cn

Yu Yao
School of Traffic and Transportation, Beijing Jiaotong University
Beijing 100044, China
Email: yaoyu1@bjtu.edu.cn

Xin Zhang
School of Traffic and Transportation, Beijing Jiaotong University
Beijing 100044, China
Email: 13114243@bjtu.edu.cn

Xuesong Zhou*
School of Sustainable Engineering and the Built Environment, Arizona State University
Tempe, AZ 85281, USA
Email: xzhou74@asu.edu

(* Corresponding authors)

Abstract

The cyclic train timetabling problem aims to synchronize limited operational resources toward a master periodic schedule of transport services. By introducing an extended time-space network construct, this paper proposes a new type of integer programming model reformulation for the cyclic train timetabling problem on a double-track railway corridor at the macroscopic level. This reformulation method also holds the promises to be applied in a broader set of routing and scheduling problems with periodic activity requirements. We also hope that this space-time network extension technique, as a special version of variable splitting methods in the dual decomposition literature, could potentially bridge the modeling gaps between cyclic and non-cyclic timetables. Specifically, the existing mathematical programming model for the periodic event scheduling problem (PESP) is transformed into a multi-commodity network flow model with two coupled schedule networks and side track capacity constraints. In addition, two dual decomposition methods including Lagrangian relaxation and Alternating Direction Method of Multipliers (ADMM), are adopted to dualize the side track capacity constraints. For each train-specific sub-problem in an iterative primal and dual optimization framework, we develop an enhanced version of forward dynamic programming to find the time-dependent least cost master schedule across the time-space network over multiple periods. ADMM-motivated heuristic methods with adjusted penalty parameters are also developed to obtain good upper bound solutions. Based on real-world instances from the Beijing-Shanghai high-speed railway corridor, we compare the numerical performance between the proposed reformulation and the PESP model that involves the standard optimization solver.

Keywords: Cyclic train timetabling; extended time-space network; Lagrangian relaxation; ADMM

1. Introduction

Railroad serves an important role in transporting long-distance passengers and massive goods using an economically attractive and environmentally friendly manner. As the passenger and freight demand for railroad transportation, in different parts of regional markets (e.g. US, China), has grown greatly in recent years (US Bureau of Transportation Statistics; Chinese National Bureau of Statistics), the railroad companies need to operate trains efficiently with limited infrastructure capacity, particularly through holistic optimization of train timetables and real time dispatching methods. The train timetabling problem has attracted much attention from researchers around the world, and we refer readers to Assad (1980), Cordeau et al. (1998), Huisman et al. (2005), Caprara et al. (2007), Harrod (2012) and Caimi et al. (2017) for excellent surveys on the train timetabling problems, while a majority of the efforts has focused on the category of non-cyclic train timetabling problems. For example, Heydar et al. (2013) reported that about one-hundred published papers involved the non-cyclic train timetabling problem, while the number dropped to less than one third for the cyclic train timetabling problem. In addition, most of the cyclic train timetabling models are built within the periodic event scheduling problem (PESP) modeling framework proposed in Serafini and Ukovich (1989), and modulo variables are usually required to deal with the non-trivial periodicity constraints (Harrod, 2012; Caimi et al., 2017). In our research, we hope our proposed method could potentially bridge the modeling gaps between cyclic and non-cyclic train timetables, by introducing a time-space network extension technique to deal with the periodicity constraints without using modulo variables, as a special version of variable splitting methods in the dual decomposition literature.

In general, the cyclic train timetabling problem determines the event times for the arrival, departure, and passing of the trains in each visited station that should be repeated in every cycle time, and it has the advantage of regularity and it is especially convenient for passengers to use the transfer service. In addition, PESP-based model usually requires that the event times are all less than the cycle length, and binary modulo variables will typically enforce two safety constraints for each pair of incompatible events corresponding to either before or after precedence relationships (see Appendix B). The non-cyclic train timetabling problem, on the other hand, is more flexible and it is suitable to serve time-varying passenger flows. In a real world case with complex railway passenger flow characteristics and network structures, such as the Beijing-Shanghai high-speed railway corridor in China, planners have to simultaneously consider a spectrum of demand patterns, ranging from short-distance-high-density trips, as well as long-distance and seasonal travels. As a result, it is important to examine and integrate the cyclic and non-cyclic schedules to explore the trade-off relationship between regularity and flexibility of train timetable services (Robenek et al., 2017; Robenek et al., 2018; Yin et al., 2019). Even though cyclic train timetable only needs to consider fewer trains and shorter planning period than the non-cyclic train timetable, the periodicity constraints considering variable trip times and dwell times and large train departure time windows can greatly increase the difficulty in solving the PESP-based model, especially for long-distance railway corridors with heterogeneous train traffic characteristics. In this paper, we formulate a new mathematical model for the cyclic train timetabling problem based on the time-discretized time-space network modeling framework, where the resulting new integer programming model could be further extended to handle the complex hybrid cyclic train timetabling problems.

To address the difficulty in finding the time-space paths for a large number of trains, researchers (such as Caprara et al., 2002) have developed a range of efficient decomposition methods to solve the train timetabling problem. For instance, a typical rolling horizon approach has been adopted in various studies including D'Ariano and Pranzo (2009), Meng and Zhou (2011) and Zhan et al., (2016). Another widely used decomposition method is Lagrangian relaxation and the related heuristics, and this research line, represented by papers from Brännlund et al. (1998), Caprara et al. (2002), Caprara et al. (2006) and Meng and Zhou (2014), aims to dualize the "hard" track capacity constraints, so as to enable the use of efficient single-train path searching algorithms. In a very recent study by Niu et al. (2018), the authors highlight a thorny modeling issue of solution symmetry in the context of Lagrangian relaxation, and has been also systematically examined in a broader modeling framework of Branch-and-Price (Barnhart et al., 1998) for general integer programming problems.

In this research, we aim to address several closely related modeling challenges in train timetabling. Specifically, we (1) reformulate the cyclic train timetabling problem by introducing an extended time-discretized time-space network modeling framework, where the special cyclic requirement is treated through a set of variable duplication constraints between the master schedule and extended schedule across different periods. (2) we apply two dual decomposition approaches, namely, the Lagrangian relaxation and Alternating Direction Method of Multipliers (ADMM), to invoke an modified version of efficient forward dynamic programming to jointly optimize the master and period-specific schedules, (3) we also adopt a new linearization technique to transform the quadratic penalty term in ADMM and the proposed method could serve as a theoretically sound foundation for enhancing the existing one-pass priority rules within an iterative primal and dual solution search framework. Through various tests on various illustrative and real-world large instances, we show the solution quality and computational efficiency of the proposed approach, in comparison to the standard PESP model and CPLEX-based optimization results for the cyclic train timetabling problem.

This paper is organized further as follows. Section 2 provides a brief literature review on the cyclic train timetabling problem. The problem statement and notations are introduced in Section 3. In Section 4, we propose a new integer programming model based on the time-discretized time-space network, and the related reformulation methods. Section 5 presents the iterative ADMM and Lagrangian relaxation-based solution procedures, as well as heuristic methods. This is followed by numerical experiments in Section 6. Finally, Section 7 provides the concluding remarks.

2. Literature review on non-cyclic and cyclic train timetabling problems

This section briefly reviews the problem characteristics, modeling methods and solution algorithms for the optimization problems being studied.

2.1 Train timetabling problem without handling special cyclic requirements

The railway planning process is usually carried out hierarchically (Lusby et al., 2011), where the optimization results of the line planning problem at the strategic level are the input parameters for the train timetabling problem at the tactical level. The fundamental elements determined in the line planning process includes train routes, frequencies, operation zones and stop patterns (Zhou and Zhong, 2005; Qi et al., 2018). Once the line planning process is completed, a non-cyclic train timetabling problem typically aims to optimize the arrival, departure and passing times of the trains at each station with various goals. In general, basic constraints considered in the non-cyclic train timetabling problem include departure time windows, minimum and maximum running times, minimum and maximum dwell times, safety headways, and prevention of illegal overtaking in the sections. Many previous studies on the non-cyclic train timetabling problem adopt the time-discretized time-space network (Mees, 1991; Brännlund et al., 1998; Caprara et al., 2002; Caprara et al., 2006; Cacchiani et al., 2008; Cacchiani et al., 2010; Harrod, 2011, 2012; Cacchiani et al., 2012; Meng and Zhou, 2014; Liu and Zhou, 2016; Yue et al., 2016; Luan et al., 2017; Jiang et al., 2017) or time-space-state network (Zhou et al., 2017; Xu et al., 2018; Meng and Zhou, 2019) modeling methods, while some other researcher also adopted the big-*M* modeling method to linearize the "if-then" conditions for train conflicts (Higgins et al., 1996; Ghoseiri et al., 2004; Zhou and Zhong, 2005; Zhou and Zhong, 2007; Mu and Dessouky, 2011; D'Ariano et al., 2017).

In particular, the train timetabling problem, based on time-discretized time-space network, can be formulated as a multi-commodity network flow problem with side track capacity constraints (Mees, 1991; Brännlund et al., 1998; Caprara et al., 2002) Specifically, Mees (1991) and Brännlund (1998) divided the railway network into track sections or blocks, and the side track capacity constraint for each track section or block was dualized by adopting the Lagrangian relaxation method. Caprara et al. (2002) further introduced a more flexible train timetabling model for a single-track railway network with stations being split into two separate nodes and the track sections between stations were treated as arcs. In addition, Harrod (2011, 2012) firstly proposed the concept of hypergraph in the time-discretized time-space network to deal with the train conflicts on the transition at cells, by which the train timetabling problem in the microscopic level for the North America single-track railway was studied.

A nice extension to the time-discretized time-space network modeling method is to introduce the third "state" dimension, which results in the discretized time-space-state network modeling method. Zhou et al. (2017) proposed the three-dimensional space-time-speed network modeling approach to achieve the integrated optimization of train timetabling problem and train speed profile with the objective of minimizing total travel cost. In particular, they proposed the Lagrangian relaxation method to dualize the side track capacity constraints, which resulted in a set of space-time-speed shortest path finding sub-problems. Besides, Xu et al. (2018) embedded the locomotive assignment decisions in the "state" dimension of the time-space-state network, where they can optimize the train timetabling and locomotive assignment problems simultaneously. Meng and Zhou (2019) incorporated passenger demand, train service, and infrastructure and rolling stock capacity into a layered network for the integrated optimization of train service plan, and passenger carrying states of the train teams were adopted as the "state" dimension to represent the dynamic pickup and drop-off of passenger groups. Another widely used approach for non-cyclic train timetabling problem is the big-*M* modeling method. Generally, the sequence of two trains is unknown at their commonly used infrastructure resources, and thus big-*M* modeling method needs to model the "if-then" conflicting relationship between two trains by introducing a sufficiently larger number big-*M* (Higgins et al., 1996). Constraint programming (Oliveira and Smith, 2000) and discrete event model (Dorfman and Medanic, 2004) were also applied to deal with the non-cyclic train timetabling problem.

2.2 Cyclic train timetabling problem

Scheduling problems with cyclic requirements have a wide range of real-life applications, such as the cyclic job shop scheduling problem (Bożejko and Wodecki, 2018), periodic vehicle routing and inventory vehicle routing problems (Mor and Speranza, 2018), road traffic signal optimization (Li et al., 2015), course timetabling problem (Socha et al., 2002), service network design problem for the road-rail intermodal freight transport in North-America railroads (Crainic and Laporte, 1997; Macharis and Bontekoning, 2004) and cyclic train timetabling problem. The cyclic train timetabling problem not only contains all of the basic constraints in the non-cyclic train timetabling problem, but also typically requires that a given frequency of train services should be repeated in each cycle (Peeters, 2003; Kroon and Peeters, 2003; Caimi et al., 2017). In addition, train regularity plays a more important role in the cyclic train timetabling problem. Since the introduction of the periodic event scheduling problem (PESP) in Serafini and Ukovich (1989), this seminal work has been served as the modeling foundation for many variants of cyclic train timetabling problems (Huisman et al., 2005; Harrod, 2012; Caimi et al., 2017). Inspired by the PESP modeling framework, many researchers have performed in-depth studies for the cyclic train timetabling problem (Odijk, 1996; Lindner, 2000; Peeters, 2003; Liebchen, 2008; Goerigk and Schöbel, 2013; Zhang and Nie, 2016; Herrigel et al., 2018), and the PESP model has been successfully applied to the European railway market (Caimi et al., 2017), such as the Dutch railway (Kroon et al., 2009) and the Berlin Underground (Liebchen, 2006, 2008). Generally, periodic event-activity network (PEAN) is constructed first to denote the constraints to be addressed in the PESP-based model, such as the regularity requirements, and modulo variables are needed to map two connected events in PEAN into the same cycle (Peeters, 2003). In addition, Peeters (2003) proposed an equivalent cycle periodicity formulation (CPF) for the cyclic train timetabling problem based on the special structure of PEAN.

With respect to the complexity of the PESP model, Liebchen (2008) provided MAXSNP-hardness proofs for two variants of the PESP model. There are a wide spectrum of solution algorithms for PESP, including the constraint generation algorithm (Odijk, 1996), genetic algorithm (Nachtigall and Voget, 1996), branch-and-bound method (Lindner, 2000), application of SAT solvers (Großmann et al., 2012; Kümmling et al., 2015; Gatterman et al., 2016) and modulo simplex method (Nachtigall and Opitz, 2008; Siebert and Goerigk, 2013; Goerigk and Schöbel, 2013). In particular, Liebchen (2004) studied a kind of cyclic train timetable with symmetry property to speed-up the CPLEX solving process. Mathias (2008) developed integer programming formulations by introducing the time discretization technique into the PESP model. Recently, Herrigel et al. (2018) proposed heuristic train grouping strategies and solved the resulted CPF models sequentially.

Over the past few decades, several important aspects of the cyclic train timetable were addressed based on the PESP model. The most intuitive one is to increase the flexibility of cyclic train timetable by allowing variable trip times. Both Kroon and Peeters (2003) and Liebchen and Möhring (2007) proposed a method to split the trip arcs into shorter ones, so that the trip time differences of trains can be reduced and illegal overtaking in the section were prevented. On the other hand, Zhang and Nie (2016) introduced a non-collision constraint that enforced the sum of binary modulo variables of running activities and safety/regularity activities for two trains in the same direction at two successive stations be equal to twice of the sum of two binary auxiliary variables, and thus the sum of the associated binary modulo variables always equaled to 0, 2, or 4 which can effectively prevent the illegal overtaking of two trains in the section. Yan and Goverde (2017) further improved the non-collision constraint in Zhang and Nie (2016) by replacing these two binary auxiliary variables with one integer auxiliary variable, which can only take the values of 0, 1 and 2.

Since the design of cyclic train timetables focuses on providing convenient services for passengers, some researchers incorporated passenger related costs into the timetable planning process. Specifically, Nachtigall and Voget (1996) designed a genetic algorithm for the PESP model which aimed to minimize the passenger waiting times for transfer. Liebchen (2006, 2008) introduced the first optimized cyclic train timetable for Berlin Underground with guaranteed maximum passenger transfer waiting times and reduced maximum train dwell times. Cordone and Redaelli (2011) integrated the passenger discrete-choice model with CPF model so that the total passenger demand captured by the trains was maximized. Siebert and Goerigk (2013) considered two methods to generate PEAN for the public transit network, namely, frequency as attribute (FA) and frequency as multiplicity (FM). The passenger paths were explicitly handled in their PESP models, which showed that passenger travel times can be improved by using their extended PESP models. Burggraeve et al. (2017) designed line planning module and cyclic train timetabling module to iteratively update the line plan and cyclic train timetable, where the line planning module aimed to optimize the passenger and operation costs and the cyclic train timetabling module was used to generate robust timetables. Very recently, Robenek et al. (2017, 2018) studied the hybrid cyclic train timetable, where the regularity of cyclic train timetable and flexibility of non-cyclic train timetable are nicely combined to improve the passenger satisfaction. Moreover, Yin et al. (2019) developed a mixed-integer programming model based on the PESP model to generate a hybrid and demand-responsive cyclic train timetable, which aimed to minimize the sum of weighted total train travel time and total passenger waiting time considering rolling stock circulation and other practical constraints. Meanwhile, they also proposed a set of performance indicators to systemically evaluate the operation efficiency and periodicity of the hybrid cyclic train timetable. In addition, there is also a trend to extend the scope of cyclic train timetabling with network planning, line planning and rolling stock/vehicle scheduling (Liebchen and Möhring, 2007; Kroon et al., 2013; Burggraeve et al., 2017; Yin et al., 2019) to obtain system-level benefits.

Another two interesting aspects for cyclic train timetabling are capacity analysis and delay management. Instead of optimizing the cyclic train timetable with fixed cycle length, Heydar et al. (2013), Petering et al. (2015), Zhang and Nie (2016) and Sparing and Goverde (2017) set the cycle length as the objective to be minimized. This unique method provides a new perspective to analyze the railway capacity with a range of factors. In addition, the delay management problem (Schöbel, 2007; Liebchen et al., 2010; Schachtebeck and Schöbel, 2010) and stable/robust cyclic train timetabling problem (Sparing and Goverde, 2017; Yan and Goverde, 2017; Yan et al., 2019) were also studied by some researchers to improve the train service reliability. More recently, Yan et al. (2019) proposed a new multi-objective optimization approach based on the PESP modeling framework to comprehensively improve the quality of the cyclic train timetable from four aspects, including train journey time, train regularity deviation, timetable vulnerability and number of overtakings. The delay management problem typically makes the wait-depart decisions on whether the connection train/vehicle should depart on time or wait for the delayed feeder train/vehicle so that the passenger delay can be reduced. Moreover, the stable/robust cyclic train timetabling problem could shed more light on improving the delay resistance ability of the train timetable in case of primary train delays.

Apart from the above previous studies that solved the cyclic train timetabling problem based on the PESP model, two studies also tried to deal with the periodicity constraints under the time-discretized time-space network modeling framework. Specifically, Caprara et al. (2002) assumed that the same train timetable would be repeated every day, and a modulo operation, which is similar to the PESP model with the cycle length of one day, was adopted such that the difference between the planned times of any two nodes/events can be correctly calculated. Therefore, trains can cross the boundary between two days and all of the potential conflicts can be captured. Furthermore, Bešinović et al. (2016) proposed a novel time-space path-based integer programming model for solving the cyclic train timetabling problem at the macroscopic level. In particular, an efficient randomized multi-start greedy heuristic algorithm was developed to iteratively minimize the total costs related to the cyclic train timetable, including the sum of train cancellation cost, train running time, dwell time and connection time extension cost, and robustness cost. Moreover, at each iteration of the greedy heuristic algorithm, a dynamic programming recursion approach with embedded modulo operation was adopted to find the least cost time-space paths of all lines in a random line-by-line order. In this study, we present another attempt to model the cyclic

timetabling problem based on the extended time-discretized time-space network without using modulo variables/operation, and two dual decomposition approaches are also developed to solve the resulting integer programming model.

Table 1 summarizes the recent studies on the non-cyclic and cyclic train timetabling problems, with three major highlights. First, most previous work on the cyclic train timetabling problem is based on the PESP model, and researchers mainly focus on solving cyclic train timetabling problem at the macroscopic level (i.e., a railway network with stations and sections). Second, the time-discretized time-space network modeling approach, through the use of Lagrangian relaxation, is effective for solving real-life non-cyclic train timetabling problem, at both macroscopic and microscopic levels. Finally, the research gap between computationally efficient time-space network-based algorithm and theoretically challenging cyclic timetabling problems can be observed.

Table 1
Summary of problem characteristics, modeling approaches and solution algorithms for non-cyclic and cyclic train timetabling problems

Publication	Planning period	Modeling approach	Objective	Solution algorithm	Variable trip time	Largest instance solved
Brännlund et al. (1998)	Non-cyclic	TSN	Max total profit	LR		A double-track railway line with 17 stations, 30 trains
Caprara et al. (2002)	Non-cyclic	TSN	Max total profit	LR	√	A single-track railway line with 39 stations, 500 trains
Cacchiani et al. (2008)	Non-cyclic	TSN	Max total profit	Column generation		A single-track railway line with up to 102 stations, 221 trains
Harrod (2011)	Non-cyclic	TSN	Max total utility	CPLEX		An 86.4 km double-track mainline with 100 trains
Meng and Zhou (2014)	Non-cyclic	TSN	Min total deviation time	LR	√	A cell-node-based railway network with 85 nodes and 97 cells, 40 trains
Jiang et al. (2017)	Non-cyclic	TSN	Max total profit	LR		A double-track railway line with 23 stations, 304 + 83 trains
Meng and Zhou (2019)	Non-cyclic	TSSN	Max total profit	LR	√	A microscopic rail network with 1032 nodes and 1614 cells, 100 trains
Odijk (1996)	Cyclic	PESP	Feasible timetable	Constraint generation		A railway station with 6 platforms, 12 trains
Nachtigall and Voget (1996)	Cyclic	PESP	Min passenger waiting time	Genetic algorithm		A railway network with 26 lines and 37 stations
Lindner (2000)	Cyclic	PESP	Min total cost	Branch-and-bound		A railway network with 297 nodes, 384 edges, and 89 lines
Kroon and Peeters (2003)	Cyclic	PESP	General	CADANS solver	√	Dutch railway network with 250 trains in one hour
Liebchen (2008)	Cyclic	PESP	Min total train idle time	CPLEX + heuristic		A 144 km subway network with 19 transfer stations
Goerigk and Schöbel (2013)	Cyclic	PESP	Min total slack time	Modulo simplex	√	A railway network with 134 lines and 319 stations
Heydar et al. (2013)	Cyclic	PESP	Min cycle time and local train dwell time	CPLEX		A single-track railway line with 70 intermediate stations
Bešinović et al. (2016)	Cyclic	TSN	Min total cost	Randomized multi-start greedy heuristic	√	A part of the Dutch railway network with 15 macroscopic timetable points (e. g., stations and stops), 25 lines and 50 trains
Zhang and Nie (2016)	Cyclic	PESP	Min cycle time	CPLEX + heuristic	√	A double-track railway line with 23 stations, 18 trains
Robenek et al. (2017)	Hybrid cyclic	PESP	Max passenger satisfaction	Simulated annealing		A railway network with 47 stations and 34 lines, 388 trains
Sparing and Goverde (2017)	Cyclic	PESP	Min cycle time	CPLEX + heuristic	√	A railway corridor with 18 train lines and 13 stations
Herrigel et al. (2018)	Cyclic	PESP	Min total cost	CPLEX + heuristic	√	A railway network with 186 operation points and 142 trains
This paper	Cyclic	TSN	Min total journey time	LR and ADMM	√	A double-track railway line with 23 stations, 36 trains

Note: TSN represents time-space network; TSSN represents time-space-state network; LR represents Lagrangian relaxation.

3. Problem statement and notations

In this paper, we study the cyclic train timetabling problem on a double-track railway corridor network $N = (V, E)$ at the macroscopic level, and the goal is to minimize the total journey times of all trains. Fig. 1 illustrates the relationship between the key concepts of this study.

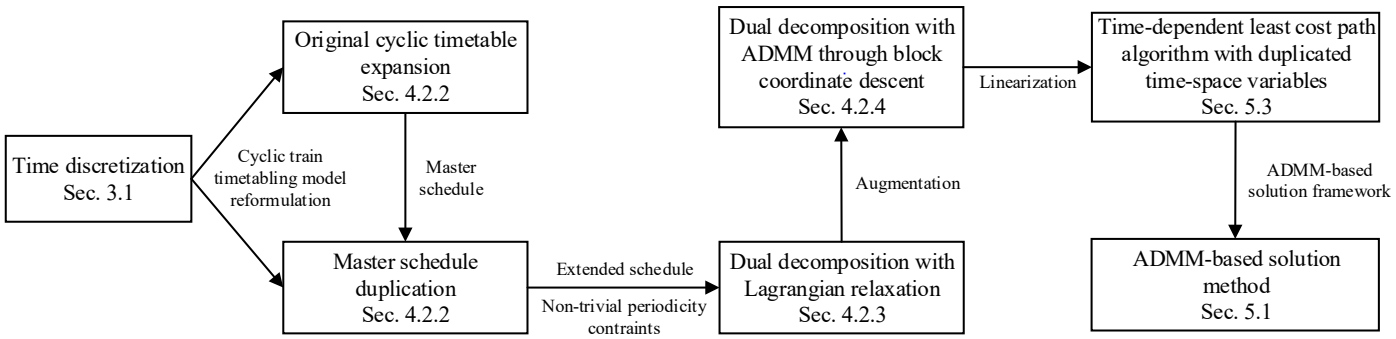
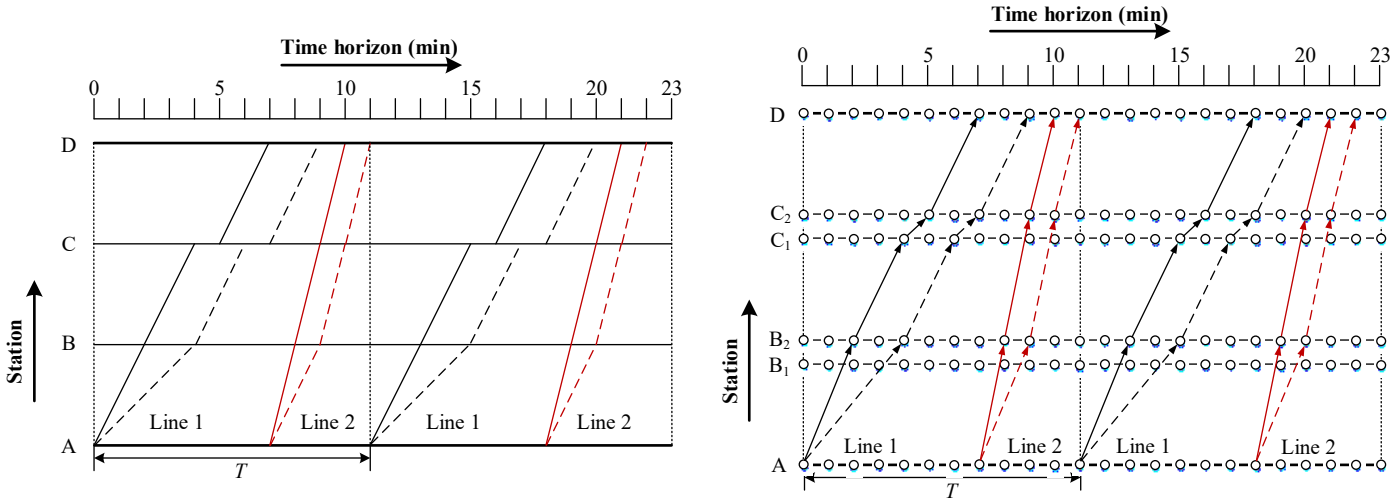


Fig. 1. Relationship between the key concepts of this study

The set V consists of all of the stations in the railway network, and the set E contains the sections that connect two adjacent stations. The cyclic train timetabling problem is to schedule a set of lines $l \in L$ periodically for every cycle length T, and each line l is associated with frequency f_l, stop pattern and operation zones. The frequency f_l requires that f_l identical trains are scheduled with an even time interval within the cycle T, which is called as the train regularity requirement. Specifically, the time intervals between the arrival or departure times of any two trains belonging to the same line are the multiple of $\lfloor T/f_l \rfloor$. In this study, for simplicity, we do not allow the flexibility on the train regularity requirement that has been considered in Zhang and Nie (2016) from the capacity analysis perspective. In addition, the train running times in the sections and dwell times in the stations are restricted with the corresponding minimum and maximum values, and train acceleration and deceleration times can be accurately considered within the minimum and maximum running times when the train stop pattern is given. The consideration of minimum and maximum running times as well as acceleration and deceleration times could lead to large differences on the actual trip times between two trains running in the same section. Hence, in order to prevent the illegal overtaking of two trains in the sections, we adopt a similar method of splitting trip arcs in Kroon and Peeters (2003) and Liebchen and Möhring (2007), where a dummy station is inserted in the middle of the section that has long distance for possible illegal overtaking resulting in two dummy sections. In this case, the acceleration and deceleration times are added to the minimum and maximum running times of the first and second dummy sections, respectively. The seven types of safety headway requirements considered in this paper are illustrated in detail in Appendix A, where arrival and departure times of two trains are separated according to those safety headway requirements. Note that those safety headway requirements are adopted by mainly considering the safety rules for the high-speed railway corridor in China.



(a) Cyclic train timetable with two lines

(b) Cyclic train timetable in the time-discretized time-space network

Fig. 2. Expression of the cyclic train timetable with variable trip times in the time-discretized time-space network

Fig. 2 shows an example of the cyclic train timetable for a double-track railway corridor with two cycles. The railway corridor contains four stations (i.e., stations A, B, C and D), and two lines (i.e., lines 1 and 2) with both frequencies are set to 1, and the cycle length equals to 11 min. In particular, line 1 in the dark color has lower speed and it has a scheduled stop at station C for 1 min, while line 2 in the red color has a higher speed and it runs from stations A to D without any stop. In addition, we consider possible variations of trip times for lines 1 and 2 in section A-B, where their trip times are within the ranges $[2, 3]$ and $[1, 2]$ respectively. Hence, two possible schedules are generated for each line, which is denoted by the solid and dash lines in Fig. 2(a) respectively. In order to express the cyclic train timetabling problem within the time-discretized time-space network modeling framework, an equivalent cyclic train diagram with discretized time intervals is depicted in Fig. 2(b). Specifically, we adopt the method in Caprara et al. (2002) where each intermediate station is split into two dummy stations, and a dummy section is naturally formed between these two dummy

stations to describe variable train dwell times. For instance, station B in Fig. 2(a) is split into two dummy stations B_1 and B_2 in Fig. 2(b). The process of constructing the time-space network can be illustrated by taking the time-space path of line 1 in Fig. 2(b) as an example. Since line 1 has no scheduled stop at station B, a directed time-space arc is connected from station A to dummy station B_2 denoting that the train travels from station A to station B. Note that, line 1 has a scheduled stop at station C, and dummy station B_2 is connected to dummy station C_1 first, and another time-space arc connects dummy stations C_1 and C_2 denoting that the train stops at station C for 1 min. By constructing the time-discretized time-space network for each train, the cyclic train timetabling problem is now transformed into a multi-commodity flow problem with side track capacity constraints and the proof on the model reformulation equivalency is given in Sec. 4.

Table 2 lists the general indices, sets, parameters and decision variables used in this paper to formulate the integrated optimization model.

Table 2
Sets, indices, parameters and variables

Indices	Definition
i, i', j, j'	Index of stations, $i, i', j, j' \in V$
(i, i')	Index of sections, $(i, i') \in E$
l	Index of lines
a, a'	Index of trains
t, t', t''	Index of the time intervals in the master time-space network, $t, t', t'' \in T$
τ, τ', τ''	Index of the time intervals in the extended time-space network, $\tau, \tau', \tau'' \in T'$
(i, t)	Index of vertexes in the master time-space network, $(i, t) \in V'$
(i, i', t, t')	Index of arcs in the master time-space network, $(i, i', t, t') \in E'$
(i, τ)	Index of vertexes in the extended time-space network, $(i, \tau) \in V''$
(i, i', τ, τ')	Index of arcs in the extended time-space network, $(i, i', \tau, \tau') \in E''$
Sets	
V	Set of stations, including the dummy stations
E	Set of sections, including the dummy sections
L	Set of lines
A	Set of trains
A_l	Set of trains belonging to line l
V'	Set of vertexes in the master time-space network
V''	Set of vertexes in the extended time-space network
V'_a	Set of vertexes in the master time-space network associated with train a
V''_a	Set of vertexes in the extended time-space network associated with train a
E'	Set of arcs in the master time-space network
E''	Set of arcs in the extended time-space network
E'_a	Set of arcs in the master time-space network associated with train a
E''_a	Set of arcs in the extended time-space network associated with train a
T	Cycle length of the cyclic train timetable
T_m	Set of discretized time intervals in the master time-space network
T_e	Set of discretized time intervals in the extended time-space network
$U(j, j', t'')$	Set of arcs in the master time-space network that are incompatible on section (j, j') at time t'', and the set of trains contained in $U(j, j', t'')$ is influenced by the settings of headway safety constraints
$U'(j, j', \tau'')$	Set of arcs in the extended time-space network that are incompatible on cell (j, j') at time τ'', and the set of trains contained in $U'(j, j', \tau'')$ is influenced by the settings of headway safety constraints
Parameters	
O_a	Origin station of train a
D_a	Destination station of train a
$[start_a, end_a]$	Departure time window of train a from its origin station O_a
$h_{dd}, h_{aa}, h_{ap}, h_{pp}, h_{pd}, h_{pa}, h_{dp}$	Safety headway between the two trains at the same station
$c_a(i, i', t, t')$	Travel cost of the time-space arc (i, i', t, t') for train a
ρ	Penalty parameter in ADMM, $\rho > 0$
f_l	Frequency of train line l
w_l	The first train with the earliest departure time in line l
H	Number of cycles for the master time-space network, where $T_m = H \cdot T$ and $T_e = 2H \cdot T$ for a given value of H
β	Integer parameter to specify to number of copies of master schedule in the extended schedule, $\beta \in \{0, \cdots, H\}$
$q_{l,a}$	Integer parameter to specify the order of train a in line l, $q_{l,a} \in \{0, \cdots, f_l - 1\}$
Variables	
$x_a(i, i', t, t')$	0-1 time-space arc selection variable for the master schedule, = 1 if train a is assigned on time-space arc (i, i', t, t') ; = 0 otherwise
$y_a(i, i', \tau, \tau')$	0-1 time-space arc selection variable for the extended schedule, = 1 if train a is assigned on time-space arc (i, i', τ, τ') ; = 0 otherwise
$\lambda_{j,j',\tau}$	Lagrangian multipliers in associated with the track capacity constraints in the extended schedule

4. Mathematical modeling

4.1 Modeling assumptions

Before introducing our proposed train timetabling models, we present some key modeling assumptions to facilitate the modeling process.

(1) One minute is set as the minimum time interval in the time-discretized time-space network, and shorter time interval values can be also applied if necessary.

(2) For trains belonging to the same line, their time-space trajectories are evenly distributed according to the time interval as a function of the cycle length divided by line frequency.

(3) The double-track railway corridor is composed of a set of interconnected stations and segments at the macroscopic level, and the safety headway requirements for the double-track railway corridor in China are adopted to ensure the safe headway between any two trains.

(4) The physical length of trains are not considered, so we model trains as single objects moving in the double-track railway corridor, and consider only trains running in one direction.

(5) Only those trains running in a single direction are considered so that those trains have no interaction with other trains running in the opposite direction.

4.2 Cyclic train timetabling model

4.2.1 General time-space network model for train timetabling problem (**M1**)

Objective function:

$$\min Z_1 = \sum_{a \in A} \sum_{(i,i',t,t') \in E'_a} c_a(i,i',t,t') \cdot x_a(i,i',t,t') \tag{1}$$

Subject to:

Flow balance constraint:

$$\sum_{i,t:(i,i',t,t') \in E'_a} x_a(i,i',t,t') - \sum_{i,t:(i',i,t',t) \in E'_a} x_a(i',i,t',t) = \begin{cases} -1 & i' = O_a, t' = start_a \\ 1 & i' = D_a, t' = T \\ 0 & otherwise \end{cases}, \quad \forall a \in A \tag{2}$$

Track capacity constraint:

$$\sum_{a \in A} \sum_{(i,i',t,t') \in U(j,j',t'')} x_a(i,i',t,t') \le 1, \quad \forall (j,j') \in E, t'' \in T \tag{3}$$

Decision variables:

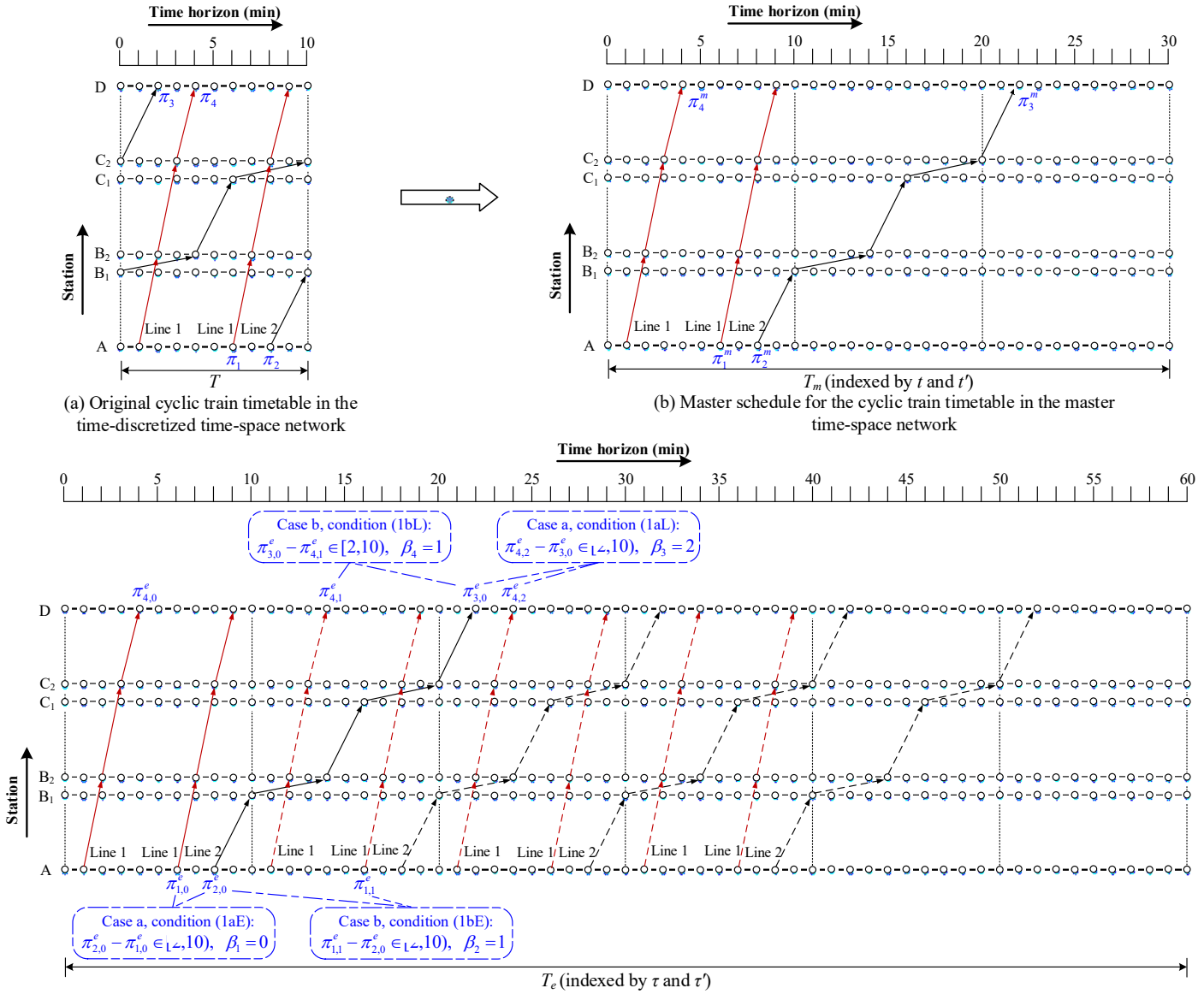
$$x_a(i,i',t,t') \in \{0,1\}, \quad \forall a \in A, (i,i',t,t') \in E' \tag{4}$$

Model **M1** is a standard binary integer programming model for the non-cyclic train timetabling problem based on the time-discretized time-space network, and its mathematical form is essentially the same as the one in Caprara et al. (2002) except for the defined objective function. The objective function in Eq. (1) is minimizing the total travel cost of all trains, and many optimization goals can be represented by the travel costs, such as the train journey times and energy consumptions. In this paper, the travel costs of the time-space arcs are specified as the corresponding train travel times such that model **M1** aims to minimize the total train journey times. Constraint (2) corresponds to the flow balance relationship, ensuring that one unique time-space path is selected by each individual train. It should be noted that the feasible arc set E'_a for train a is used instead of original arc set E' in constraint (2), so that the time-space arcs that train a can travel through can be customized to reduce the number of arcs to be examined, and the pre-specified train stop requirements can also be honored accordingly. Constraint (3) is the track capacity constraint, which describes the time-space resource usage constraints according to the safety headway requirements in Appendix A, where at most one single train can occupy one of the time-space arcs in the clique $U(j,j',t)$. Finally, constraint (4) specifies the domain of time-space arc selection variables.

4.2.2 Cyclic train timetabling model based on extended time-space network (**M2**)

Now we proceed to the steps of timetable expansion and duplications illustrated in Fig. 1. The non-cyclic train timetabling model **M1** can be used to generate the master schedule in the master time-space network. For the PESP model, modulo variables are introduced to map all train time-space paths into the same cycle (Peeters, 2003). Fig. 3 (a) shows an example of the cyclic train timetable which contains two lines, where line 1 in the red color has a frequency of 2 and line 2 in the dark color has a frequency of 1. It can be seen that the cycle length equals to 10 min and the time-space path for the train of line 2 crosses the cyclic boundary T, where part of the time-space path that is located outside of the cycle and it is mapped to the be beginning of the cyclic train timetable. Inspired by the work in Heydar et al. (2013), we propose the concept of the master schedule, which expands the planning horizon for H times of T resulting a master time-space network with the length of the planning horizon T_m be equal to $H \cdot T$. The value of H

is set to the minimum possible integer value such that T_m is greater than the maximum possible arrival times of all trains at their destination stations.



(a) Original cyclic train timetable in the time-discretized time-space network

(b) Master schedule for the cyclic train timetable in the master time-space network

(c) Extended schedule for the cyclic train timetable in the extended time-space network

Fig. 3. Transformation of the original cyclic train timetable into the master and extended schedules

Moreover, Fig. 3 (b) shows the master schedule in the master time-space network corresponding to the cyclic train timetable in Fig. 3(a). The maximum possible arrival time of the trains is equal to 22 min in Fig. 3(b), so the value of H is set to 3 and T_m is equal to 30 min. Note that even though time-space paths of the trains can travel cross the cyclic boundaries in the master schedule, it is required that the departure times of all trains are located within the range $[0, T)$. In addition, the departure time window $[start_a, end_a]$ for the first train a of a line $l \in L$ is set to $[0, \min\{\lfloor T/f_l \rfloor, T-1\}]$, and other trains in the same line are evenly distributed with the time interval $\lfloor T/f_l \rfloor$. For instance, the first and second trains of line 1 departs from station A at times 1 and 6 min respectively with an time interval of $\min\{\lfloor 10/2 \rfloor, 10-1\} = 5$ min.

The time-space paths of the trains in the master schedule in the extended time-space network in Fig.3 (b) are now moving forward in the time direction without looping back, similar to the handling of trains in a non-cyclic train timetable. To further guarantee that there is no conflict in the cyclic train timetable, we propose the idea of duplicating the master schedule to multiple copies (i.e., extended schedules) in the frame of the extended time-space network.

The length of the planning horizon T_e for the extended time-space network is equal to $2H \cdot T$, and the trains in the master schedule are copied by $H+1$ times and each time has the offset of T. Fig. 3 (c) shows an example of the extended schedule for the master schedule in Fig. 3(b), and the value of T_e is equal to 60 min, and the train time-space paths are copied by 4 times. Obviously and conceptually, it can be seen that there is no conflict, such as illegal overtaking within the section, over the whole planning horizon of the extended time-space network. Hence, it can be guaranteed that the original cyclic train timetable is feasible. The master and

extended schedules are coupled together, where the master schedule aims to use a similar manner of planning non-cyclic train timetable to schedule trains and the extended schedule is to resolve the train conflicts in the cyclic train timetable. We will proceed to **Proposition 1** at the end of Section 4.2.2 for a formal proof.

Mathematically, two sets of time-space arc selection variables $x_a(i,i',t,t')$ and $y_a(i,i',\tau,\tau')$ are designed for the master and extended schedules respectively. Note that time intervals in the master time-space network are indexed by t and t' while time intervals in the extended time-space network are indexed by τ and τ'. In model **M2**, the objective function (5) aims to minimize the total journey times of all trains in the extended schedule. Constraint (6) is the flow balance constraint that finds one unique time-space path in the master time-space network for the first train w_l of each line $l \in L$, where $x_{w_l}(i,i',t,t')$ in constraint (6) represents the binary time-space arc selection variable for the first train w_l. Constraint (7) generates the time-space path for the rest trains in the same line $l \in L$ by shifting the time-space path of the first train w_l with an offset of $\min\{\lfloor T/f_l \rfloor, T-1\}$. The integer parameter $q_{l,a}$ in constraint (7) specifies the order of train a in line l, and $q_{l,a}$ is restricted to the set $\{1,\cdots,f_l-1\}$ such that the first train w_l in line l is excluded from constraint (7).

One of key contributions in our proposed reformulation is the variable splitting/duplication technique. Constraint (8) is the consensus constraint between the master schedule and extended schedule, which simply duplicates the time-space paths of the trains in the master schedule by $H+1$ times in the extended schedule. Note that the integer parameter β in constraint (8) enforces the number of copies of master schedule in the extended schedule, and it belongs to the set $\{0,\cdots,H\}$. The form of constraint (8) is similar to the nonanticipativity constraint, across the first-stage and second-stage decisions of the two-stage stochastic mixed integer programming (SMIP) problems, in Crainic et al. (2011) and Boland et al. (2018) that has been well tackled by using the progressive hedging approach to solve the two-stage SMIP problems. Note that, in our proposed model, because constraint (8) only performs the duplication operation, there is no need to dualize it into the objective function. Constraint (9) is the track capacity constraint enforcing that there is no conflict in the entire planning horizon of the extended time-space network. Constraints (10) and (11) define the domain of binary time-space arc selection variables for the master and extended schedules.

$$\min Z_2 = \sum_{a\in A}\sum_{(i,i',\tau,\tau')\in E_a''} c_a(i,i',\tau,\tau')\cdot y_a(i,i',\tau,\tau') \tag{5}$$

Subject to:

Flow balance constraint:

$$\sum_{i,t:(i,i',t,t')\in E'_{w_l}} x_{w_l}(i,i',t,t') - \sum_{i,t:(i',i,t',t)\in E'_{w_l}} x_{w_l}(i',i,t',t) = \begin{cases} -1 & i'=O_{w_l}, t'=start_a \\ 1 & i'=D_{w_l}, t'=T_m \\ 0 & otherwise \end{cases}, \quad \forall l\in L \tag{6}$$

Master schedule coupling constraint:

$$x_a\big(i,i',t+q_{l,a}\cdot\min\{\lfloor T/f_l\rfloor,T-1\},t'+q_{l,a}\cdot\min\{\lfloor T/f_l\rfloor,T-1\}\big) = x_{w_l}(i,i',t,t'),\\ \forall l\in L, a\in A_l, (i,i',t,t')\in E', f_l>1, q_{l,a}\in\{1,\cdots,f_l-1\} \tag{7}$$

Extended schedule duplication constraint:

$$y_a(i,i',\tau,\tau') = x_a(i,i',t+\beta T,t'+\beta T),\\ \forall a\in A,(i,i',t,t')\in E',(i,i',\tau,\tau')\in E'',\beta\in\{0,\cdots,H\},\tau=t+\beta T,\tau'=t'+\beta T \tag{8}$$

Track capacity constraint:

$$\sum_{a\in A}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') \le 1, \quad \forall (j,j')\in E, \tau''\in T_e \tag{9}$$

Domain of variables:

$$x_a(i,i',t,t')\in\{0,1\}, \quad \forall a\in A,(i,i',t,t')\in E' \tag{10}$$

$$y_a(i,i',\tau,\tau')\in\{0,1\}, \quad \forall a\in A,(i,i',\tau,\tau')\in E'' \tag{11}$$

Proposition 1. A feasible extended schedule in the extended time-space network can always correspond to a feasible cyclic train timetable, and vice versa.

Proof. Through the duplication operation in constraint (8), there are $H+1$ copies of the time-space path of each train in the extended time-space network which are evenly distributed by a time interval of T. Since $H\cdot T$ is larger than the maximum possible arrival times of all trains to their destinations in the master schedule and the time-space paths of all trains in the master schedule are duplicated by $H+1$ times, it is always possible to map a pair of time-space arcs of the master schedule into the same cycle in the extended schedule. More importantly, for the time-space arcs in the H^{th} cycle, i.e., the last cycle of the master schedule, all the other time-space arcs located before the H^{th} cycle can be shifted to the same H^{th} cycle with at most $(H-1)\cdot T$ offsets. Therefore, a cyclic train timetable that has an identical form to the original cyclic train timetable can always be obtained in the H^{th} cycle of the extended

schedule. Fig. 3 shows the transformation of the original cyclic train timetable into the extended schedule, where the train time-space paths in the 3rd cycle of Fig. 3(c) is equivalent to the original cyclic train timetable in Fig. 3(a). We next show how the designed duplication operation in the extended schedule, just as modulo variables in the PESP model, has the same ability in handling the safety constraints of the cyclic train timetable.

Suppose k and k' are two different events and activity (k, k') belongs to the set of safety activities $\mathcal{A}_{\text{safe}}$ in the PEAN of the PESP model, $\pi_k, \pi_{k'} \in [0, T)$ are integer variables representing the planned times of events k and k', $z_{k',k}$ denotes the binary modulo variable where $z_{k',k} = 1$ if $\pi_{k'} > \pi_k$ and $z_{k',k} = 0$ otherwise, and $h_{k,k'}$ is the minimum headway between events k and k'. In the PESP model, safety constraint $\pi_k - \pi_{k'} + z_{k',k}T \in [h_{k,k'}, T - h_{k,k'}]$ enforces the feasibility of the cyclic train timetable without conflicts. If we suppose $\pi_k < \pi_{k'}$, the two safety constraints can be simplified as $\pi_k - \pi_{k'} + T \geq h_{k,k'}$ with $z_{k',k} = 1$ and $\pi_{k'} - \pi_k \geq h_{k,k'}$ with $z_{k,k'} = 0$, which should be satisfied simultaneously in the cyclic train timetable.

Without loss of generality, the planned times of event k and k' in the *master* schedule are denoted by π_k^m and $\pi_{k'}^m$, and π_k^m could still be smaller than $\pi_{k'}^m$ or vice versa. Moreover, the duplicated events for events k and k' with the planned times of $\pi_{k,\beta}^e = \pi_k^m + \beta T, \beta \in [0, H]$ and $\pi_{k',\beta}^e = \pi_{k'}^m + \beta T, \beta \in [0, H]$ are generated in the *extended* schedule, respectively. Table 3 lists the correspondence relationship of the safety constraints between the extended schedule and PESP model with $\pi_k < \pi_{k'}$, and two cases corresponding to the two safety constraints can be stated as follows:

(1) If both conditions (1aE) and (1aL) are satisfied in the extended schedule, safety constraints (0a) is enforced in the PESP model

Conditions (E) implies that $\pi_{k,0}^e$ is earlier than $\pi_{k',0}^e$, i.e., $\pi_{k,0}^e < \pi_{k',0}^e$, while condition (L) means $\pi_{k,0}^e$ is later than $\pi_{k',0}^e$, i.e., $\pi_{k,0}^e > \pi_{k',0}^e$. Condition (1aE) specifies that there is a largest integer $\beta_1 \in [0, H]$ such that $\pi_{k',0}^e - \pi_{k,\beta_1}^e \in [h_{k,k'}, T)$ for $\pi_{k,0}^e < \pi_{k',0}^e$ in the extended schedule, which implies that the absolute difference between $\pi_{k',0}^e$ and π_{k,β_1}^e is less than T and greater than or equal to the minimum headway $h_{k,k'}$. Similarly, condition (1aL) requires that there is a smallest integer $\beta_3 \in [0, H]$ such that $\pi_{k',\beta_3}^e - \pi_{k,0}^e \in [h_{k,k'}, T)$ for $\pi_{k,0}^e > \pi_{k',0}^e$. Fig. 3 shows four events 1, 2, 3 and 4 with planned times of $\pi_1 = 6, \pi_2 = 8, \pi_3 = 2$ and $\pi_4 = 4$, where $\pi_1^m = 6$, $\pi_2^m = 8$, $\pi_3^m = 22$ and $\pi_4^m = 4$ in the master schedule and $\pi_{1,0}^e = 6$, $\pi_{1,1}^e = 16$, $\pi_{2,0}^e = 8$, $\pi_{3,0}^e = 22$, $\pi_{4,0}^e = 4$, $\pi_{4,1}^e = 14$, and $\pi_{4,2}^e = 24$ in the extended schedule. In addition, the headway $h_{k,k'}$ is set to 2 min and cycle length T equals to 10 min. Therefore, $\pi_{2,0}^e - \pi_{1,0}^e \in [2,10)$ with $\beta_1 = 0$ and $\pi_{4,2}^e - \pi_{3,0}^e \in [2,10)$ with $\beta_3 = 2$ are identical to $\pi_2 - \pi_1 \geq 2$ and $\pi_4 - \pi_3 \geq 2$, respectively.

(2) If both conditions (1bE) and (1bL) are satisfied in the extended schedule, safety constraints (0b) is enforced in the PESP model

For conditions (1bE) and (1bL), there are a smallest integer $\beta_2 \in [0, H]$ such that $\pi_{k,\beta_2}^e - \pi_{k',0}^e \in [h_{k,k'}, T)$ for $\pi_{k,0}^e < \pi_{k',0}^e$ and a largest integer $\beta_4 \in [0, H]$ such that $\pi_{k,0}^e - \pi_{k',\beta_4}^e \in [h_{k,k'}, T)$ for $\pi_{k,0}^e > \pi_{k',0}^e$. In Fig. 3, it can be shown that $\pi_{1,1}^e - \pi_{2,0}^e \in [2,10)$ with $\beta_2 = 1$ and $\pi_{3,0}^e - \pi_{4,1}^e \in [2,10)$ with $\beta_4 = 1$ are identical to $\pi_1 - \pi_2 + 10 \geq 2$ and $\pi_3 - \pi_4 + 10 \geq 2$, respectively.

In short, an extended schedule with $H + 1$ copies of the master schedule actually generates $(H + 1) \times (H + 1)$ precedence constraints between each pair of events in the original cyclic train timetable, where two integers β_1 and β_2 for condition (E) or β_3 and β_4 for condition (L) within the range $[0, H]$ can always be found corresponding to the precedence relationships in constraints (0a) and (0b), respectively. In addition, model **M2** can avoid the usage of binary sequence variables in the big-*M* modeling method to enforce the necessary headway requirements. Furthermore, compared to the time-space network formulation in Caprara et al. (2002) that does not include multiple copies of the schedule, model **M2** using the time-space network extension technique can guarantee a conflict-free cyclic train timetable even when the train time-space path traverses across more than 2 cycles. On the other hand, a master schedule which is actually equivalent to an extended schedule with only one copy cannot guarantee that all of the four conditions in Table 3 are satisfied. Thus, an extended schedule is sufficient for ensuring a conflict-free cyclic train timetable.

Table 3

Correspondence relationship of the safety constraints between the extended schedule and PESP model with $\pi_k < \pi_{k'}$

Safety constraints for PESP model		Condition (E)	Condition (L)
		$\pi_{k,0}^e < \pi_{k',0}^e$ for extended schedule	$\pi_{k,0}^e > \pi_{k',0}^e$ for extended schedule
Case a	$\pi_{k'} - \pi_k \geq h_{k,k'}$ (0a)	$\beta_1 = \underset{\beta_1 \in [0,H]}{\arg\max}\{\pi_{k',0}^e - \pi_{k,\beta_1}^e \in [h_{k,k'}, T)\}$ (1aE)	$\beta_3 = \underset{\beta_3 \in [0,H]}{\arg\min}\left\{\pi_{k',\beta_3}^e - \pi_{k,0}^e \in [h_{k,k'}, T)\right\}$ (1aL)
Case b	$\pi_k - \pi_{k'} + T \geq h_{k,k'}$ (0b)	$\beta_2 = \underset{\beta_2 \in [0,H]}{\arg\min}\{\pi_{k,\beta_2}^e - \pi_{k',0}^e \in [h_{k,k'}, T)\}$ (1bE)	$\beta_4 = \underset{\beta_4 \in [0,H]}{\arg\max}\left\{\pi_{k,0}^e - \pi_{k',\beta_4}^e \in [h_{k,k'}, T)\right\}$ (1bL)

In addition, the objective function (5) minimizes the total travel costs of all trains in the extended schedule which is equivalent to the optimization of the master schedule. Because the master schedule is the expansion of the original cyclic train timetable, and the cyclic train timetable is also optimized. A detailed comparison of model **M2** with the PESP-based cyclic train timetabling model in Zhang and Nie (2016) which has the modified objective of minimizing the total train journey times is given in Appendix B. The comparison results show that model **M2** also has the ability to handle the same practical constraints except the relaxation of train regularity, especially the non-collision and flexible overtaking constrains. In short, optimizing extended schedules in the extended time-space network is equivalent to the optimization process of cyclic train timetables.

4.2.3 Dual decomposition of model **M2** with Lagrangian relaxation (**M3**)

We now move the two dual decomposition steps illustrated in Fig. 1. By using the Lagrangian relaxation method, the side track capacity constraint (9) of Model **M2** can be dualized into the objective function after introducing the Lagrangian multiplier $\lambda_{j,j',\tau''}$, resulting in a new model **M3**. The new objective function (12) has an additional linear penalty term for the violation of side track capacity constraint, and constraint (13) restricts the value of $\lambda_{j,j',\tau''}$ to be larger than 0.

Objective function:

$$\min Z_3 = Z_2 + \sum_{(j,j')\in E}\sum_{\tau''\in T_e}\lambda_{j,j',\tau''}\left[\sum_{a\in A}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') - 1\right] \tag{12}$$

Subject to:

Constraints (6)-(8), (10)-(11) and (13).

$$\lambda_{j,j',\tau''} > 0, \qquad \forall (j,j') \in E, \tau'' \in T_e \tag{13}$$

Model **M3** consists of a set of train-specific sub-problems with the form of objective function in Eq. (14), which can be solved by finding the least cost time-dependent shortest path for each train in the extended schedule with the time-space arc usage cost $\gamma_a(i,i',\tau,\tau')$ in Eq. (15). In addition, Eq. (16) specifies the usage cost $\gamma_a^m(i,i',t,t')$ of the time-space arc (i,i',t,t') in the master schedule according to the duplication relationship between the extended and master schedules in Eq. (8). The values of Lagrangian multipliers $\lambda_{j,j',\tau''}^{k+1}$ and step size α^k at the k^{th} iteration are updated in Eqs. (17) and (18), respectively. It should be noted that Eq. (17) adopts the classical subgradient method to update the values of Lagrangian multipliers $\lambda_{j,j',\tau''}^{k+1}$, which has also been applied by some previous research papers with discretized time-space network modeling framework, such as the work by Brännlund et al. (1998), Caprara et al. (2002) and Meng and Zhou (2014).

$$\begin{aligned} Z_3^a &= \sum_{(i,i',\tau,\tau')\in E_a''} c_a(i,i',\tau,\tau')\cdot y_a(i,i',\tau,\tau') + \sum_{(j,j')\in E}\sum_{\tau''\in T_e}\lambda_{j,j',\tau''}\left[\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') - 1\right] \\ &= \sum_{(i,i',\tau,\tau')\in E_a''} \gamma_a(i,i',\tau,\tau')\cdot y_a(i,i',\tau,\tau') - \sum_{(j,j')\in E}\sum_{\tau''\in T_e}\lambda_{j,j',\tau''} \end{aligned} \tag{14}$$

$$\gamma_a(i,i',\tau,\tau') = c_a(i,i',\tau,\tau') + \sum_{(j,j')\in E}\sum_{\tau''\in T_e:(i,i',\tau,\tau')\in U'(j,j',\tau'')}\lambda_{j,j',\tau''}, \qquad \forall (i,i',\tau,\tau') \in E_a'' \tag{15}$$

$$\gamma_a^m(i,i',t,t') = \sum_{\beta=0}^{H}\gamma_a(i,i',t+\beta\cdot T, t'+\beta\cdot T), \qquad \forall a\in A, (i,i',t,t')\in E'_{w_l} \tag{16}$$

$$\lambda_{j,j',\tau''}^{k+1} = \max\left\{0, \lambda_{j,j',\tau''}^{k} + \alpha^k\left[\sum_{a\in A}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') - 1\right]\right\}, \qquad \forall (j,j')\in E, \tau''\in T_e \tag{17}$$

$$\alpha^k = 1/(k+1) \tag{18}$$

4.2.4 Dual decomposition of model **M2** with ADMM through block coordinate descent (**M4**)

Several useful dual decomposition approaches have been developed to reduce the complexities in solving various linear and non-linear optimization problems (Bertsekas, 1999; Boyd et al., 2011). Particularly, three typical dual decomposition techniques are the Lagrangian relaxation (Fisher, 1981), Augmented Lagrangian relaxation (Fortin and Glowinski, 2000) and ADMM (Glowinski and Marroco, 1975; Gabay and Mercier, 1976; Boyd et al., 2011). Lagrangian relaxation is a dual ascent method and it can dualize the "hard" constraints into the objective function, and the original complicated problem can be decomposed into many easier and independent sub-problems. Augmented Lagrangian relaxation further introduces an extra quadratic penalty term into the objective function, in an attempt to improve the robustness and functional convexity of the Lagrangian relaxation. However, the quadratic penalty term has made the variables coupled with each other, and it is difficult to update the variables in a parallel way and typically requires quadratic programming solvers. By combing the Augmented Lagrangian relaxation with the block coordinate descent method (Bertsekas, 1999), ADMM was developed where variables can be updated sequentially in a block-by-block manner. As a result, ADMM has the inherent advantages of breaking symmetry and strong convexity while maintaining good problem decomposition structure. Table 4 shows the detailed comparison between the above three dual decomposition approaches. In addition, an illustrative example of ADMM with three blocks is provided in Appendix C. The theoretical convergence of applying ADMM in integer programs is a very subtle issue, detailed discussions can be found at a very recent study by Boland et al. (2018) and a recent working paper by Yao et al. (2019).

Table 4
Characteristics of three dual decomposition approaches

Approaches	Quadratic penalty terms	Robust	Decomposable	Fully parallel implementation
Lagrangian relaxation	No	No	Yes	Yes
Augmented Lagrangian relaxation	Yes	Yes	No	No
ADMM	Yes	Yes	Yes	No

In our specific timetabling application, ADMM also dualizes the side track capacity constraint (9) with a quadratic penalty term be introduced into the objective function (19) of model **M4** except for the linear penalty term of Lagrangian relaxation. Moreover, Boyd et al. (2011), Crainic et al. (2011) and a recent study by Yao et al. (2019) applied ADMM or Augmented Lagrangian relaxation to dualize the complicating equality constraints, such as the constraint (C2) of the illustrative example in Appendix C. In order to show how to complicating inequality constraint (9) from a theoretical perspective, a continuous and non-negative slack variable $s_{j,j',\tau''} \in [0,1]$ is introduced to transform constraint (9) into an equality constraint, i.e., $\sum_{a\in A}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + s_{j,j',\tau''} - 1 = 0$. The maximum value of $s_{j,j',\tau''}$ is set to 1 as the term $\sum_{a\in A}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')$ can be equal to 0 for any clique $U'(j,j',\tau'')$, and constraint (20) specifies the range of values of $s_{j,j',\tau''}$. Therefore, the quadratic term in Eq. (19) turns to the form of $\left\|\sum_{a\in A}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + s_{j,j',\tau''} - 1\right\|_2^2$ after introducing the slack variable $s_{j,j',\tau''}$. Models **M4** and **M3** share the same constraints, but the value of Lagrangian multiplier $\lambda_{j,j',\tau''}$ is updated in Eq. (21) where the step size α is replaced by the penalty parameter ρ. It should be noted that since model **M3** can also directly handle the complicating inequality constraint (9) with the requirements of $\lambda_{j,j',\tau''} > 0$ and $\lambda_{j,j',\tau''}$ is updated by Eq. (21), the slack variable $s_{j,j',\tau''}$ is not needed to be introduced into the term Z_3 of Eq. (19). In addition, even though the mathematical form of Eq. (21) is similar to Eq. (17) that utilizes the subgradient method, the dual update step size ρ used in Eq. (21) will lead to quite different performance between ADMM and the classical subgradient method. In particular, the penalty parameter ρ is directly incorporated into the quadratic penalty term in Eq. (19) to regulate the scale of penalty for violating the side track capacity constraints. Moreover, the value of penalty parameter ρ is adjusted in a different manner compared to the adjustment of step size α in Eq. (18). The value of penalty parameter ρ is specifically increased adaptively according to the violation of track capacity constraints at each iteration which will be introduced in detail in Sec. 5.1.

Objective function:

$$\min Z_4 = Z_3 + \frac{\rho}{2}\sum_{(j,j')\in E}\sum_{\tau''\in T_e}\left\|\sum_{a\in A}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + s_{j,j',\tau''} - 1\right\|_2^2 \tag{19}$$

Subject to:

Constraints (6)-(8), (10)-(11), (13) and (20).

$$0 \le s_{j,j',\tau''} \le 1, \qquad \forall (j,j') \in E, \tau'' \in T_e \tag{20}$$

$$\lambda_{j,j',\tau''}^{k+1} = \max\left\{0, \lambda_{j,j',\tau''}^{k} + \rho\left[\sum_{a\in A}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') - 1\right]\right\}, \qquad \forall (j,j') \in E, \tau'' \in T_e \tag{21}$$

An important property for applying ADMM to integer programming programs is that, when only binary variables are involved in sub-problems with linear and separable structure, it is possible to linearize the quadratic penalty term in ADMM. Interested readers can refer to Yao et al. (2019) for the handling of binary variables in ADMM for the vehicle routing application. In addition, this technique has also been applied by Crainic et al. (2011) in the context of two-stage SMIP formulation after relaxing the nonanticipativity constraint through Augmented Lagrangian relaxation , where the linearization could be performed as the relaxed nonanticipativity constraint only consisted of binary variables. In the timetabling problem under consideration in this study, model **M4** involves only binary time-space arc selection variables, the quadratic penalty term in objective function (19) can be also reduced to a fully linearized objective function in the time-space least cost path search. In the following, we consider how to first tackle the challenges in the quadratic function (19) and then apply the block coordinate descent method to solve the sub-problem for each train.

Due to the rolling update scheme of ADMM, only one train is optimized each time with all the other trains' time-space paths being fixed. If the current train is denoted by a, the rest trains belong to the set $A' = A/\{a\}$. Note that another important feature for the linearization of quadratic penalty term in Yao et al. (2019) is that the dualized constraint can be transformed into a constraint with only binary variables that are in associated with the current train a. We use the constant $n_a(j,j',\tau'')$, and its short hand notation, n_a, in Eq. (22) to represent the number of trains in set A' that occupy the discretized time-space resources of arc (j,j') at time τ''. By substituting n_a into the quadratic term $\left\|\sum_{a\in A}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + s_{j,j',\tau''} - 1\right\|_2^2$ of Eq. (19), it can be linearized as in Eq. (23). Note that the transformation from $\left[\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')\right]^2$ into $\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')$ in Eq. (23) is due to the fact that train a can chose at most one of the time-space arcs (i,i',τ,τ') in the set $U'(j,j',\tau'')$. Thus, at most one of the binary time-space arc selection variables $y_a(i,i',\tau,\tau')$ of train a in the set $U'(j,j',\tau'')$ is equal to 1.

$$n_a = \sum_{a'\in A'}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_{a'}(i,i',\tau,\tau'), \qquad \forall (j,j') \in E, \tau'' \in T_e \tag{22}$$

$$
\begin{aligned}
&\left\| \sum_{a\in A} \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + s_{j,j',\tau''} - 1 \right\|_2^2 \\
&= \left\| \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + n_a + s_{j,j',\tau''} - 1 \right\|_2^2 \\
&= \left[\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') \right]^2 + 2 \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')\left(n_a + s_{j,j',\tau''} - 1\right) + \left(n_a + s_{j,j',\tau''} - 1\right)^2 \\
&= \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + 2 \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')\left(n_a + s_{j,j',\tau''} - 1\right) + (n_a + s_{j,j',\tau''} - 1)^2 \\
&= \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')\left(2n_a + 2s_{j,j',\tau''} - 1\right) + (n_a + s_{j,j',\tau''} - 1)^2
\end{aligned} \tag{23}
$$

In order to determine the value of $s_{j,j',\tau''}$, a function $L_{s_{j,j',\tau''}}$ in Eq. (24) where $s_{j,j',\tau''}$ is the only variable can be derived according to Eq. (23). The symbol Q in Eq. (24) denotes the items in Eq. (23) that are irrelevant to $s_{j,j',\tau''}$. Since $s_{j,j',\tau''}$ is a continuous variable, the theoretical global minimum value of $L_{s_{j,j',\tau''}}$ can be obtained at the point $s^*_{j,j',\tau''}$ in Eq. (25). However, two cases shall be considered to calculate the actual feasible value of $s^*_{j,j',\tau''}$ according to the value of n_a and the condition that $s_{j,j',\tau''} \in [0,1]$. First, when $n_a \geq 1$, the theoretical value of $s^*_{j,j',\tau''}$ is less than or equal to 0. Moreover, it can be shown that the value of $L_{s_{j,j',\tau''}}$ will monotonically increase within the range $[0,1]$ with $n_a \geq 1$. Therefore, the optimum and feasible value of $s^*_{j,j',\tau''}$ equals to 0, and Eq. (25) can be transformed into $\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')(2n_a - 1) + (n_a - 1)^2$ by setting $s_{j,j',\tau''} = 0$. Second, when $n_a = 0$, $s^*_{j,j',\tau''}$ equals to $1 - \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')$ whose value belongs to the range $[0,1]$. Thus, the value of quadratic term in Eq. (23) is equal to 0 after substituting $n_a = 0$ and $s_{j,j',\tau''} = 1 - \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')$ into Eq. (23). The linearized results of the quadratic term in Eq. (23) for the two cases are expressed in Eq. (26).

$$
L_{s_{j,j',\tau''}} = s^2_{j,j',\tau''} + 2s_{j,j',\tau''} \left[\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + n_a - 1 \right] + Q \tag{24}
$$

$$
s^*_{j,j',\tau''} = 1 - \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') - n_a \tag{25}
$$

$$
\left\| \sum_{a\in A} \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + s_{j,j',\tau''} - 1 \right\|_2^2 = \begin{cases} \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')(2n_a - 1) + (n_a - 1)^2 & n_a \geq 1 \\ 0 & n_a = 0 \end{cases} \tag{26}
$$

For the first case of $n_a \geq 1$ with $\left\| \sum_{a\in A} \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + s_{j,j',\tau''} - 1 \right\|_2^2 = \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')(2n_a - 1) + (n_a - 1)^2$, we can also derive a set of train-specific sub-problems with the new objective function Z^a_4 in Eq. (27). The usage cost $\gamma'_a(i,i',\tau,\tau')$ of time-space arc (i,i',τ,τ') in the extended schedule equals to $c_a(i,i',\tau,\tau') + \sum_{(j,j')\in E} \sum_{\tau''\in T_e:(i,i',\tau,\tau')\in U'(j,j',\tau'')} \left[\lambda_{j,j',\tau''} + \rho(2n_a - 1)/2\right]$. For the second case of $n_a = 0$ with $\left\| \sum_{a\in A} \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') + s_{j,j',\tau''} - 1 \right\|_2^2 = 0$, there are no quadratic penalty terms in Eq. (27) such that $Z^a_4 = \sum_{(i,i',\tau,\tau')\in E''_a} \gamma'_a(i,i',\tau,\tau') \cdot y_a(i,i',\tau,\tau') - \sum_{(j,j')\in E} \sum_{\tau''\in T_e} \lambda_{j,j',\tau''}$, where $\gamma'_a(i,i',\tau,\tau') = c_a(i,i',\tau,\tau') + \sum_{(j,j')\in E} \sum_{\tau''\in T_e:(i,i',\tau,\tau')\in U'(j,j',\tau'')} \lambda_{j,j',\tau''}$. In short, the theoretical and unified form of $\gamma'_a(i,i',\tau,\tau')$ for the two cases can be expressed as $\gamma'_a(i,i',\tau,\tau') = c_a(i,i',\tau,\tau') + \sum_{(j,j')\in E} \sum_{\tau''\in T_e:(i,i',\tau,\tau')\in U'(j,j',\tau'')} \left\{\lambda_{j,j',\tau''} + \max\{0, \rho(2n_a - 1)/2\}\right\}$ in Eq. (28). Note that since the linearized arc usage cost $\gamma'_a(i,i',\tau,\tau')$ in Eq. (28) does not contain the slack variable $s_{j,j',\tau''}$, constraint (20) is not needed to be considered in model **M4**.

$$
\begin{aligned}
Z^a_4 &= \sum_{(i,i',\tau,\tau')\in E''_a} c_a(i,i',\tau,\tau') \cdot y_a(i,i',\tau,\tau') + \sum_{(j,j')\in E} \sum_{\tau''\in T_e} \lambda_{j,j',\tau''} \left[\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') - 1 \right] \\
&+ \frac{\rho}{2} \sum_{(j,j')\in E} \sum_{\tau''\in T_e} \left[\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau')(2n_a - 1) + (n_a - 1)^2 \right] \\
&= \sum_{(i,i',\tau,\tau')\in E''_a} c_a(i,i',\tau,\tau') \cdot y_a(i,i',\tau,\tau') + \sum_{(j,j')\in E} \sum_{\tau''\in T_e} \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') \left[\lambda_{j,j',\tau''} + \frac{\rho}{2}(2n_a - 1)\right] \\
&+ \frac{\rho}{2} \sum_{(j,j')\in E} \sum_{\tau''\in T_e} (n_a - 1)^2 - \sum_{(j,j')\in E} \sum_{\tau''\in T_e} \lambda_{j,j',\tau''} \\
&= \sum_{(i,i',\tau,\tau')\in E''_a} \gamma'_a(i,i',\tau,\tau') \cdot y_a(i,i',\tau,\tau') + \frac{\rho}{2} \sum_{(j,j')\in E} \sum_{\tau''\in T_e} (n_a - 1)^2 - \sum_{(j,j')\in E} \sum_{\tau''\in T_e} \lambda_{j,j',\tau''}
\end{aligned} \tag{27}
$$

$$\gamma'_a(i,i',\tau,\tau') = \begin{cases} c_a(i,i',\tau,\tau') + \sum_{(j,j')\in E}\sum_{\tau''\in T_e:(i,i',\tau,\tau')\in U'(j,j',\tau'')}\left[\lambda_{j,j',\tau''} + \frac{\rho}{2}(2n_a-1)\right] & n_a \geq 1 \\ c_a(i,i',\tau,\tau') + \sum_{(j,j')\in E}\sum_{\tau''\in T_e:(i,i',\tau,\tau')\in U'(j,j',\tau'')}\lambda_{j,j',\tau''} & n_a = 0' \end{cases}$$
$$\forall (i,i',\tau,\tau') \in E''_a \tag{28}$$
$$= c_a(i,i',\tau,\tau') + \sum_{(j,j')\in E}\sum_{\tau''\in T_e:(i,i',\tau,\tau')\in U'(j,j',\tau'')}\left\{\lambda_{j,j',\tau''} + \max\left\{0, \frac{\rho}{2}(2n_a-1)\right\}\right\}, \quad \forall (i,i',\tau,\tau') \in E''_a$$

However, our numerical experiments indicate that the revised form of $\gamma'_a(i,i',\tau,\tau')$ in Eq. (29) can lead to better performance of ADMM. Eq. (29) is equivalent to Eq. (28) when $n_a \geq 1$, while Eq. (29) may reduce the value of $\gamma'_a(i,i',\tau,\tau')$ when $n_a = 0$ compared to Eq. (28), which implies that the current train a is more likely to be attracted to use the time-space arcs with no conflicts. Note that Larsson and Yuan (2004), which utilized the disaggregate simplicial decomposition method to generate the linearized arc costs for the Frank-Wolfe type sub-problems within the Augmented Lagrangian relaxation-based solution framework, also adopted a similar form of linearized arc usage cost to the one in Eq. (29) for solving a large-scale multicommodity routing problem. Moreover, the usage cost $\gamma_a^{m'}(i,i',t,t')$ of time-space arc (i,i',t,t') in the master schedule is determined in Eq. (30). It can be shown that the original non-linear objective function (19) with a quadratic penalty term is linearized by adding a term of $\sum_{(j,j')\in E,\tau''\in T_e}\sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')}\max\{0,\lambda_{j,j',\tau''} + \rho(2n_a-1)/2\}$ into the (linear) time-space arc cost, which is one of the critical differences between the Lagrangian relaxation and ADMM-based dual decomposition methods. As a result, an efficient forward dynamic programming approach can be directly applied to solve each train-specific sub-problem of model **M4**. Note that each train can be viewed as a block in ADMM, and the block coordinate descent property of ADMM (Bertsekas, 1999) requires the arc usage costs $\gamma'_a(i,i',\tau,\tau')$ in Eq. (29) need to be updated as soon as the time-dependent least cost path searching process for one train is finished. Consequently, the potential impact of the time-space paths of other trains except the current train will be embedded into the time-dependent least cost path searching process. In Appendix D, a hypothetic example is used to illustrate models **M3** and **M4** with detailed calculation steps, where the potential symmetry issue in the Lagrangian relaxation and symmetry breaking mechanism in ADMM are explained in details.

$$\gamma'_a(i,i',\tau,\tau') = c_a(i,i',\tau,\tau') + \sum_{(j,j')\in E}\sum_{\tau''\in T_e:(i,i',\tau,\tau')\in U'(j,j',\tau'')}\max\left\{0,\lambda_{j,j',\tau''} + \frac{\rho}{2}(2n_a-1)\right\}, \quad \forall (i,i',\tau,\tau') \in E''_a \tag{29}$$

$$\gamma_a^{m'}(i,i',t,t') = \sum_{\beta=0}^{H}\gamma'_a(i,i',t+\beta\cdot T, t'+\beta\cdot T), \quad \forall a\in A, (i,i',t,t')\in E'_{w_l} \tag{30}$$

5. Solution methods

This section aims to introduce the algorithmic details of two solution methods. The first one is the ADMM-based solution method in Sec. 5.1 for model **M4**, including the solution framework and the iterative algorithmic steps for the generation of upper bound and lower bound solutions. The second one is the Lagrangian relaxation-based solution method in Sec. 5.2 for model **M3**, which has been illustrated in detail in the previous work by Brännlund et al. (1998), Caprara et al. (2002) and Meng and Zhou (2014), and thus we only introduce how to apply a priority rule-based heuristic to generate the upper bound solutions. Lastly, the time-dependent least cost path algorithm with duplicated time-space variables to solve each train-specific sub-problem in models **M3** and **M4** is illustrated in detail in Sec. 5.3.

5.1 ADMM-based solution method

Fig. 4 provides the conceptual illustration on the ADMM-based solution framework for iteratively updating of the best lower and upper bounds. Some symbol definitions for the ADMM-based solution procedure are listed in Table 5, and the detailed solution procedure of ADMM is then provided in **Algorithm 1 (ADMM)**.

The solution procedure of **Algorithm 1 (ADMM)** consists of six steps. In Step 1, the initial values of iteration number k, penalty parameter ρ, incremental amount of penalty parameter ε, Lagrangian multipliers $\lambda^k_{j,j',\tau''}$, arc usage costs $\gamma_a(i,i',\tau,\tau')$ and $\gamma'_a(i,i',\tau,\tau')$, best lower bound value $Best\_LB^*$ and best upper bound value $Best\_UB^*$, and sequence of lines Seq are provided. In Step 2, pure lower bound solution generation involves solving each train-specific sub-problem by using the time-dependent least cost path algorithm with duplicated time-space variables (cf. **Algorithm 3 (TDLCP)**) in Sec. 5.3, where the time-space arc usage costs in the extended schedule are set to $\gamma_a(i,i',\tau,\tau')$ in Eq. (15). Note that the quadratic penalty term is not considered when generating the dual solution, which is due to the infeasibility of the dual solution and large penalty values for the violations of capacity constraints. Moreover, the best lower bound value $Best\_LB^*$ needs to be updated by the equation $Best\_LB^* = \max\{Best\_LB^*, LB^k\}$ for every lower bound solution with the lower bound value be equal to LB^k at the k^{th} ADMM iteration.

In Step 3 for generating the ADMM solution and updating the best upper bound value, the time-space arc usage costs in the extended schedule are set to $\gamma'_a(i,i',\tau,\tau')$ in Eq. (29) and the time-dependent least cost paths for the trains are computed in a sequential line-by-line manner using **Algorithm 3 (TDLCP)** in Sec. 5.3. In particular, Fig. 5 illustrates the rolling update scheme of ADMM in Step 3. Each row in Fig. 5 represents the update of one line each time, and the line number within the circle in each row

means the current line that is to be optimized, while the lines within the rectangle are those lines with their time-space paths temporarily being fixed at the current inner iteration. In particular, if the p^{th} line in the set Seq is to be optimized for finding the time-dependent least cost path at the $(k+1)^{th}$ ADMM iteration, the time-space paths of the lines $l_{p+1}, l_{p+2}, \cdots, l_{|L|}$ are fixed to the values at the k^{th} ADMM iteration, while the time-space paths of the lines $l_1, l_2, \cdots, l_{p-1}$ are fixed to the values at the $(k+1)^{th}$ ADMM iteration. In addition, it should be noted that arc usage costs $\gamma'_a(i,i',\tau,\tau')$ need to be updated dynamically according to Eq. (29) at each inner iteration so as to the symmetry between different lines can be broken. Furthermore, if the ADMM solution obtained at the k^{th} iteration is feasible with the upper bound value be equal to UB^k, a feasible cyclic train timetable is generated and the best upper bound value shall be updated by the equation $Best\_UB^* = \min\{Best\_UB^*, UB^k\}$. In addition, we also calculate the Lagrangian profit of each line $l \in L$ and a new line sequence Seq is obtained through sorting the lines in descending order by their Lagrangian profits (Caprara et al., 2002; Meng and Zhou, 2014).

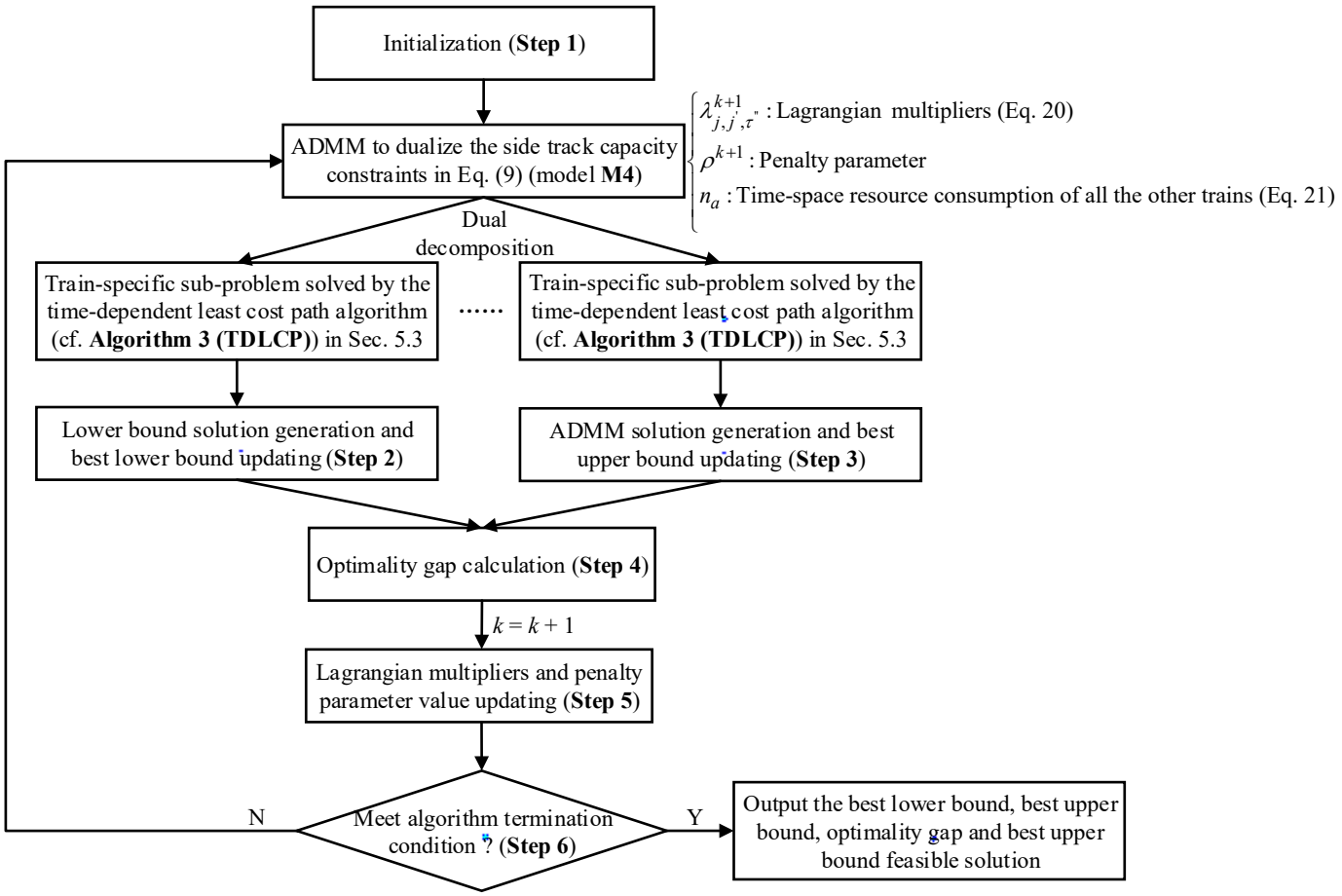


Fig. 4. ADMM-based solution framework for iteratively updating of the best lower and upper bounds (cf. **Algorithm 1 (ADMM)**)

After updating the best lower bound value $Best\_LB^*$ in Step 2 and best upper bound value $Best\_UB^*$ in Step 3, the optimality gap of ADMM GAP is calculated in Step 4 by the equation of $100\% * (Best\_UB^* - Best\_LB^*)/Best\_LB^*$. In Step 5, the values of Lagrangian multipliers $\lambda^{k+1}_{i,i',\tau''}$ are updated by Eq. (21) with the value of penalty parameter be equal to ρ^k, and a heuristic rule considering the violation of side track capacity constraints in Eq. (9) is designed to increase or retain the value of penalty parameter ρ^{k+1} at the $(k+1)^{th}$ ADMM iteration, so that good feasible solutions satisfying the primal and dual feasibility requirements can be obtained (Boyd et al., 2011; Yao et al., 2019). Finally, Step 6 determines whether to continue the iterative algorithmic process by going to Step 2 or to terminate the ADMM algorithm after reaching the maximum iteration number K.

Table 5
Symbol definitions for the ADMM-based solution method

Symbol	Definition
Seq	Sequences of lines in set L
ε	Incremental amount for increasing the value of penalty parameter ρ
μ	Weighting factor used in the condition of whether increasing the value of penalty parameter ρ or not
LB^k, UB^k	Lower bound and upper bound values of ADMM at the k^{th} iteration
$Best\_LB^*, Best\_UB^*$	Best lower bound and upper bound values of ADMM

Algorithm 1 (ADMM): iterative algorithmic steps of the ADMM-based solution method

Step 1. (Initialization)

Set iteration number $k = 0$, penalty parameter $\rho = 2$, Lagrangian multipliers $\lambda_{j,j',\tau''}^{k} = 2$, arc usage costs $\gamma_a(i,i',\tau,\tau') = c_a(i,i',\tau,\tau')$ and $\gamma'_a(i,i',\tau,\tau') = c_a(i,i',\tau,\tau')$, best lower bound $Best\_LB^* = -\infty$, best upper bound $Best\_UB^* = +\infty$, initial sequence of lines Seq, incremental amount of penalty parameter ε

Step 2. (Pure lower bound solution generation and best lower bound updating)

Update arc usage costs $\gamma_a(i,i',\tau,\tau')$ by Eq. (15)

Use **Algorithm 3 (TDLCP)** in Sec. 5.3 to find the time-dependent least cost paths for all the trains in the extended schedule

Calculate the lower bound LB^k at the k^{th} iteration and update the best lower bound $Best\_LB^* = \max\{Best\_LB^*, LB^k\}$

Step 3. (ADMM solution generation and best upper bound updating)

For each line $l \in Seq$ **do**

Update arc usage costs $\gamma'_a(i,i',\tau,\tau')$ by Eq. (29)

Use **Algorithm 3 (TDLCP)** in Sec. 5.3 to find the time-dependent least cost paths for all the trains related to line l in the extended schedule

End for each line $l \in Seq$

If the current ADMM solution is feasible

Calculate the upper bound UB^k at the k^{th} iteration and update the best upper bound $Best\_UB^* = \min\{Best\_UB^*, UB^k\}$

Calculate the Lagrangian profit for each line $l \in L$ by the ratio of dividing free-flow travel time (i.e., the minimum travel time of the train w_l without considering the side track capacity constraints) by the corresponding least cost travel time in the dual solution of the train w_l (Caprara et al., 2002; Meng and Zhou, 2014).

Sort lines in descending order by their Lagrangian profits and generate new line sequence Seq

End If

Step 4. (Optimality gap calculation)

Compute the optimality gap GAP by the equation $100\% * (Best\_UB^* - Best\_LB^*)/Best\_LB^*$

Step 5. (Lagrangian multipliers and penalty parameter value updating)

Let $k = k + 1$, update the Lagrangian multipliers $\lambda_{i,i',\tau''}^{k+1}$ by Eq. (21) with the value of penalty parameter ρ^k at the k^{th} iteration

Increase or retain the penalty parameter value ρ^{k+1} according to the following condition (Yao et al., 2019):

$$violation_{j,j',\tau''} = \max\{0, \sum_{a\in A} \sum_{(i,i',\tau,\tau')\in U'(j,j',\tau'')} y_a(i,i',\tau,\tau') - 1\}, \quad \forall (j,j') \in E, \tau'' \in T_e$$

$$\rho^{k+1} = \begin{cases} \rho^k + \varepsilon, & \sum_{(j,j')\in E} \sum_{\tau''\in T_e} violation_{j,j',\tau''}^2 \geq \mu \sum_{(j,j')\in E} \sum_{\tau''\in T_e} violation_{j,j',\tau''} \\ \rho^k, & otherwise \end{cases}$$

where $violation_{j,j',\tau''}$ is the violation of track capacity constraints in the clique $U'(j,j',\tau'')$, and ε and μ take the suitable values within the ranges $[1,5]$ and $[1,4]$ after observing the general performance of the ADMM-based solution method, respectively.

Step 6. (Algorithm termination condition)

Terminate the ADMM algorithm if k reaches the maximum iteration number K; otherwise, go to Step 2.

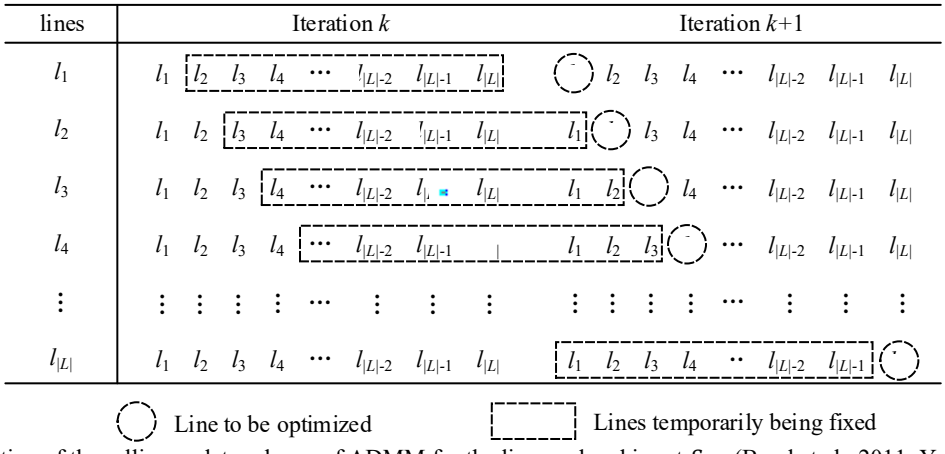


Fig. 5. Illustration of the rolling update scheme of ADMM for the lines ordered in set Seq (Boyd et al., 2011; Yao et al., 2019)

5.2 Lagrangian relaxation-based solution method

The solution procedure of Lagrangian relaxation for model **M3** is similar to that of ADMM, and readers can refer to Brännlund et al. (1998), Caprara et al. (2002) and Meng and Zhou (2014) for detailed algorithmic steps of Lagrangian relaxation for the train timetabling problem. Moreover, the upper bound feasible solution of Lagrangian relaxation can be derived by utilizing the dual information with a priority rule-based heuristic algorithm (Brännlund et al., 1998; Caprara et al., 2002; Meng and Zhou, 2014) in **Algorithm 2 (PRUB)**.

Algorithm 2 (PRUB) includes four algorithmic steps. In Step 1, the value of Lagrangian profit for each line $l \in L$ is calculated by using the same method in **Algorithm 1 (ADMM)**, and the priorities of lines are determined in Step 2 through sorting the lines in descending order by the values of Lagrangian profits of the first train w_l for each line $l \in L$. In Step 3, the time-space paths of the lines are generated sequentially starting from the line with the highest priority. In particular, the arc usage costs are set to $c_a(i, i', t, t')$, i.e., the travel time of the train on the time-space arc (i, i', t, t'). Moreover, **Algorithm 3 (TDLCP)** of Sec. 5.3 is used to find the time-dependent least cost path for the first train w_l of line l, and the time-space paths of other related trains in the master and extended schedules can be generated naturally according to constraints (7) and (8). In addition, the costs of the time-space arcs that conflict with the time-space paths of the trains associated with the current line l are set to a sufficiently large number (Meng and Zhou, 2014; Wei et al., 2017), so that necessary headway requirements in Appendix A are guaranteed. The sequential search process in Step 3 is iterated until all trains in the set of lines L are scheduled. Finally, the optimality gap of Lagrangian relaxation is updated in Step 4.

Algorithm 2 (PRUB): a priority rule-based heuristic for generating the upper bound solutions

Step 1. (Lagrangian profit calculation)
Obtain the Lagrangian profit for each line $l \in L$.

Step 2. (Line priority ranking)
Lines in set L are sorted in descending order by their Lagrangian profit values.

Step 3. (Sequential feasible time-space path generation)
Step 3.1 For the current line l with the highest priority, use **Algorithm 3 (TDLCP)** in Sec. 5.3 to find the time-dependent least cost path for the first train w_l in line l, where the arc usage costs are set to $c_a(i, i', t, t')$ in Table 2.
Step 3.2 Generate the time-space paths for other related trains by constraints (7) and (8), and mark the associated conflicting time-space arcs as infeasible to guarantee the necessary safety headways in Appendix A (Meng and Zhou, 2014; Wei et al., 2017).
Step 3.3 Repeat Steps 3.1 and 3.2 until all trains in the set of lines L are scheduled and go to Step 4.

Step 4. (Upper bound updating)
Update the best upper bound and calculate the corresponding optimality gap if the currently feasible solution has a smaller total travel cost.

5.3 Time-dependent least cost path algorithm with duplicated time-space variables

The time-dependent least cost path algorithm based on the forward dynamic programming approach was proposed by Ziliaskopoulos and Mahmassani (1993) and Chabini (1998) for the dynamic road transportation networks. The time-space paths of each train a consists of waiting, travelling and dummy arcs. Waiting arcs from (O_a, τ) to $(O_a, \tau + 1)$ allows train a to wait at the origin station until train a departs from the origin station, which must be contained within the departure time window $[start_a, end_a]$. Moreover, by adopting the station splitting method in Caprara et al. (2002), a set of travelling arcs from (i, τ) to (i', τ') can be constructed for train a with flexible travel times in the sections and dwell times in the stations. Dummy arcs from (D_a, τ) to $(D_a, \tau + 1)$ connect the actual arrival time of train a at the destination station to the end time of planning horizon. Interested readers can check more details for constructing the time-space network and the associated time-dependent least cost path algorithms in Pallottino and Scutella (1998), Caprara et al. (2002), Meng and Zhou (2014), Zhou et al. (2018) and Tong et al. (2019).

The symbol definitions needed for the time-dependent least cost algorithm are provided in Table 7, and the detailed algorithmic steps are then introduced. Eq. (31) describes the key label cost update condition in step 2 corresponding to the lines 14 to 18 of **Algorithm 3 (TDLCP)**, where the usage cost of the arc (i, i', t, t') for the first train w_l of line l in the master schedule is calculated by summing the arc usage costs of the arcs belonging to the same line l and all of the duplicated time-space arcs/variables. In addition, the forward dynamic programming in step 2 of **Algorithm 3 (TDLCP)** can obtain the time-dependent least cost path for the first train w_l, and the time-space paths of other trains in the same line l can also be generated according to Eqs. (7) and (8). Note that the arc usage cost $r'_{w_l}(i, i', \tau, \tau')$ is used in **Algorithm 3 (TDLCP)** for the ADMM-based solution method, and $\gamma'_{w_l}(i, i', \tau, \tau')$ can be replaced by $\gamma_{w_l}(i, i', \tau, \tau')$ directly for the Lagrangian relaxation-based solution method.

$$\sigma_{w_l}(i,t) + \sum_{q_{l,a}=0}^{f_l-1} \sum_{\beta=0}^{H} r'_{w_l}\left(i, i', t + q_{l,a} \cdot \lfloor T/f_l \rfloor + \beta \cdot T, t' + q_{l,a} \cdot \lfloor T/f_l \rfloor + \beta \cdot T\right) < \sigma_{w_l}(i', t'), \qquad \forall l \in L, (i, i', t, t') \in E'_{w_l} \tag{31}$$

Table 7
Symbol definitions for the time-dependent least cost algorithm with duplicated time-space variables

Symbol	Definition
$\sigma_a(i,t)$	Label cost of time-space vertex (i,t) for train a
$pred_a(i',t')$	Predecessor of time-space vertex (i',t') for train a
sum	Regular parameter for storing the sum of time-space arc usage costs

Algorithm 3 (TDLCP): time-dependent least cost path algorithm in forward dynamic programming with duplicated time-space variables

Input: Physical railway network $G = (V, E)$; cycle length T; frequency f_l of train line l; origin station O_{w_l}, destination station D_{w_l}, departure time window $[0, \min\{\lfloor T/f_l \rfloor, T-1\}]$ and revised time-space arc usage cost $\gamma'_{w_l}(i, i', \tau, \tau')$ for the first train w_l of line l in the

extended time-space network; number of cycles H for the master time-space network; length of planning period T_m and T_e for the master and extended time-space networks

Output: least cost time-space paths of each train $a \in A_l$ in line l

Step 1. (Initialization)

Set the label cost of the time-space vertex $\sigma_{w_l}(i,t) = +\infty$, the time-space vertex predecessor $pred_{w_l}(i,t) = (-1,-1)$, the label cost $\sigma_{w_l}(O_{w_l},t) = 0, \forall t = 0,\cdots,\min\{\lfloor T/f_l \rfloor, T-1\}$

Step 2. (Label updating in forward dynamic programming)

For each line $l \in L$ **do**

  For each time t = 0 to T_m **do** //search in the master schedule

    For each outgoing arc $(i,i',t,t') \in E'_{w_l}$ of vertex $(i,t) \in V'_{w_l}$ **do**

      $sum = 0$

      For each $q_{l,a} = 0$ to $f_l - 1$ and $\beta = 0$ to H **do** //calculate the sum of the arc usage costs in the extended schedule

        $\tau = t + q_{l,a} \cdot \lfloor T/f_l \rfloor + \beta \cdot T$

        $\tau' = t' + q_{l,a} \cdot \lfloor T/f_l \rfloor + \beta \cdot T$

        $sum = sum + \gamma'_{w_l}(i,i',\tau,\tau')$

      End for each $q_{l,a} = 0$ to $f_l - 1$ and $\beta = 0$ to H

      If $\sigma_{w_l}(i,t) + sum < \sigma_{w_l}(i',t')$ // update label cost and vertex predecessor

        $\sigma_{w_l}(i',t') = \sigma_{w_l}(i,t) + sum$

        $pred_{w_l}(i',t') = (i,t)$

      End If

    End for each outgoing arc $(i,i',t,t') \in E'_{w_l}$ of vertex $(i,t) \in V'_{w_l}$

  End for each time t = 0 to T_m

End for each line $l \in L$

Step 3. (Fetch the least cost time-space path for other related trains in the extended schedule for line l)

Step 3.1 Find the time-space vertex (D_{w_l}, τ^*) with the least label cost at the destination station D_{w_l}

Step 3.2 Back trace the time-space path for train w_l with respect to the time-space vertex predecessors $pred_{w_l}(i',\tau')$

Step 3.3 Generate the time-space paths for other related trains in the extended schedule by Eqs. (14) and (15)

Step 3.4 Algorithm termination.

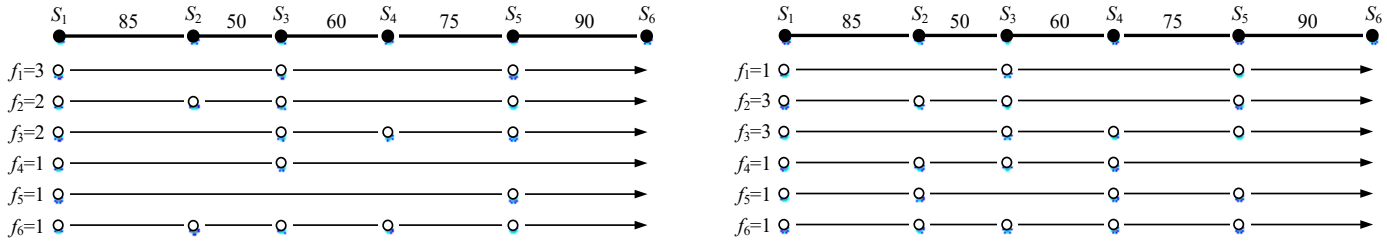
6. Numerical experiments

In this section, test cases based on the artificial and real-life railway networks are designed to test the efficiency and solution quality of our proposed models and solution algorithms. The Lagrangian relaxation (i.e., LR-M3-C++) and ADMM (i.e., ADMM-M4-C++)-based solution methods are implemented in C++ on the Visual Studio 2015 platform. Moreover, the modified PESP-based cyclic train timetabling model in Zhang and Nie (2016) (see Appendix B) with the objective of minimizing the total train journey times and fixed cycle length T is solved by CPLEX 12.8.0, MATLAB 2015b (i.e., PESP-CPLEX) to serve as the benchmarks. In addition, all of the models and algorithms are tested on a desktop computer with i7-7700 @ 3.6 GHz CPU and 32 GB RAM.

6.1 Illustrative experiments based on the artificial railway network

The artificial railway network of a double-track railway corridor with 6 stations in Fig. 6 is adapted from Zhang and Nie (2016) where Fig. 6(a) and Fig. 6(b) are corresponding to two different line plans. Fig. 6(a) shows heterogeneous train stop patterns which are adverse to the capacity usage, while trains in Fig. 6(b) have more balanced train stop patterns. Moreover, total of 6 lines are required to be scheduled in the network, and the frequency of each line is given in Fig. 6(a) and 6(b), respectively. As a result, both of the two illustrative examples contain 10 trains, and two speed grades corresponding to the fast and slow trains are assigned to those 10 trains. Besides, the distance in kilometers between every two adjacent stations is also labeled in Fig. 6 where the average distance is 72 km. The input datasets, C++ source codes and cyclic train diagram visualization tool in Python for those two illustrative examples can be downloaded from the GitHub website: https://github.com/YXZhangSWJTU/CyclicTrainTimetabling_ETSN.

Basic parameters needed for the artificial railway network are provided in Table 8. In particular, the values of six different headway retirements are given, and the minimum and maximum dwell times of the trains within the stations are restricted to the range [1, 10] min, so that overtaking can be allowed at the stations. In addition, the acceleration and deceleration times for fast and slow trains are specified as (3, 3) min and (2, 3) min, respectively. The speed range of fast and slow trains are specified as 280-300 km/h and 230-250 km/h, respectively. Specifically, the lines 1, 4 and 5 consist of fast trains and the trains in lines 2, 3 and 6 have slower speed. For the illustrative example 1, the cycle length T is set to 120 min, while the cycle length T of the illustrative example 2 is set to 90 min. Furthermore, the values of the parameter H for those two illustrative examples are set to 3, and the lengths of T_m and T_e for the illustrative example 1 are equal to $3 \times 120 = 360$ min and $2 \times 3 \times 120 = 720$ min, respectively. For the illustrative example 2, the lengths of T_m and T_e equal to $3 \times 90 = 270$ min and $2 \times 3 \times 90 = 540$ min, respectively. Besides, the number of trains in T_m is equal to the number of trains to be scheduled, while the number of trains in T_e is equal to $H + 1 = 4$ times of the number of trains in T_m. For instance, the number of trains in T_m of the illustrative example 1 with 6 lines is equal to 10, while the corresponding number of trains in T_e is equal to $4 \times 10 = 40$. Moreover, the maximum number of iterations K for Lagrangian relaxation and ADMM are set to 1000. We next perform a series of experiments based on the two illustrative examples on the number of lines and speed differences.



(a) Illustrative example 1 with heterogeneous train stop patterns (b) Illustrative example 2 with more balanced train stop patterns

Fig. 6. Two illustrative examples based on the artificial railway network in Zhang and Nie (2016)

Table 8
Basic parameters for the artificial railway network

Parameters	Values (min)
$h_{dd}, h_{aa}, h_{ap}, h_{pp}, h_{pd}, h_{pa}, h_{dp}$	5, 4, 4, 3, 3, 3, 5
Minimum and maximum dwell times in stations	[1, 10]
Acceleration and deceleration times	(3, 3) for fast trains, (2, 3) for slow trains

(1) *Different number of lines*

The experimental results for these two illustrative examples with different numbers of lines are provided in Tables 9-12. Please note that since the number of lines is usually not equal to the corresponding number of trains, the information on the number of trains is also provided in Tables 9-12 nearby the number of lines. Tables 9 and 10 include the lower bound and upper bound values, computation times and optimality gaps for LR-M3-C++, ADMM-M4-C++ and PESP-CPLEX. In particular, two types of optimality gaps are provided in Table 10 to systematically measure the quality of upper bound and lower bound solutions. The column "GAP" represents the original optimality gap regarding the relative difference between the upper bound and lower bound values of the corresponding method, while the column "BestGAP" denotes the best optimality gap whose value is calculated by adopting the best lower bound value among those three methods. Table 11 lists the detailed compositions of total computation times of Lagrangian relaxation and ADMM, including the computation times of the two main algorithmic steps for calculating the lower bound and upper bound (ADMM) solutions and other algorithmic steps. In addition, Table 12 records the number of iterations and the corresponding computation times needed for Lagrangian relaxation, ADMM and CPLEX to find the feasible and best solutions.

Five points can be drawn from the computational results in Tables 9-12. First, CPLEX can obtain optimal solutions for all of the illustrative examples within short time periods, which can provide the best lower and upper bound values. Second, the quality of lower bound and upper bound values of ADMM are slightly better than the Lagrangian relaxation which results in smaller original optimality gaps (i.e., the column "GAP") and best optimality gaps (i.e., the column "BestGAP") of ADMM. In particular, Lagrangian relaxation obtain a feasible solution for the illustrative example 1 with 6 lines with the largest original and best optimality gaps of 8.3% and 3.6% , while ADMM can achieve a feasible solution with the original and best optimality gaps be equal to 2.33% and 0%, respectively. Third, it can be seen from the best optimality gaps of ADMM in Table 10 that the upper bound values of ADMM are very close to the optimal values of CPLEX and the lower bound values of ADMM are also good enough to achieve small optimality gaps. Fourth, the computation times of Lagrangian relaxation and ADMM are proportional to the number of lines, while the computation times of CPLEX can increase dramatically. In addition, both Lagrangian relaxation and ADMM spend most of their computational efforts in calculating the lower bound and upper bound (ADMM) solutions, and these two algorithmic steps essentially involve finding the time-dependent least cost paths for the trains in the extended time-space network. Finally, Table 12 shows that all of the three methods can find their feasible and best solutions quite efficiently, while the embedded branch-and-cut algorithm of CPLEX needs much more times to prove the optimality gaps of the corresponding best solutions.

Table 9
Lower bound and upper bound values (in minutes) of two illustrative examples with a different number of lines

# of lines (trains)		LR-M3-C++		ADMM-M4-C++		PESP-CPLEX	
		Lower bound	Upper bound	Lower bound	Upper bound	Lower bound	Upper bound
Illustrative example 1	2 (5)	504.4	537	510	537	522	522
	4 (8)	811.6	858	825	852	849	849
	6 (10)	1010.2	1094	1032	1056	1056	1056
Illustrative example 2	2 (4)	432.4	436	435.6	436	436	436
	4 (8)	873.4	883	879	883	883	883
	6 (10)	1071.7	1118	1098.2	1111	1110	1110

Table 10
Computation times (in seconds) and optimality gaps of two illustrative examples with a different number of lines

# of lines (trains)	LR-M3-C++			ADMM-M4-C++			PESP-CPLEX		
	CPU time	GAP (%)	BestGAP	CPU time	GAP (%)	BestGAP	CPU time	GAP (%)	BestGAP

				(%)			(%)			(%)
Illustrative example 1	2 (5)	6.07	6.46	2.87	4.51	5.29	2.87	0.17	0	0
	4 (8)	10.34	5.72	1.06	8.3	3.27	0.35	8.8	0	0
	6 (10)	12.7	8.3	3.6	12	2.33	0	25.5	0	0
Illustrative example 2	2 (4)	4.72	0.83	0	3.38	0.09	0	0.03	0	0
	4 (8)	8.22	1.1	0	6.66	0.46	0	2.8	0	0
	6 (10)	10.31	4.32	0.72	9.64	1.17	0.09	88.31	0	0

Table 11
Computation times (in seconds) of lower bound and upper bound (ADMM) solutions of two illustrative examples with a different number of lines

# of lines (trains)		LR-M3-C++			ADMM-M4-C++		
		Lower bound solution	Upper bound solution	Other steps	Lower bound solution	ADMM solution	Other steps
Illustrative example 1	2 (5)	2.03	3.82	0.22	1.82	2.4	0.29
	4 (8)	3.77	6.38	0.19	3.34	4.58	0.38
	6 (10)	5.21	7.31	0.18	4.91	6.71	0.38
Illustrative example 2	2 (4)	1.48	3.09	0.15	1.37	1.81	0.2
	4 (8)	3.13	4.91	0.18	2.73	3.64	0.29
	6 (10)	4.19	5.99	0.13	4.02	5.31	0.31

Table 12
Feasible and best solution iterations as well as solution times (in seconds) of two illustrative examples with a different number of lines

# of lines (trains)		LR-M3-C++				ADMM-M4-C++				PESP-CPLEX	
		FeasIt	FeasTi	BestIt	BestTi	FeasIt	FeasTi	BestIt	BestTi	FeasTi	BestTi
Illustrative example 1	2 (5)	1	0.01	1	0.01	20	0.09	21	0.09	0.00	0.17
	4 (8)	11	0.11	11	0.11	2	0.02	327	2.71	0.00	0.49
	6 (10)	6	0.08	6	0.08	205	2.46	212	2.54	0.00	7.13
Illustrative example 2	2 (4)	1	0.00	1	0.00	1	0.00	1	0.00	0.00	0.03
	4 (8)	3	0.02	12	0.1	6	0.04	15	0.1	0.00	0.01
	6 (10)	794	8.19	794	8.19	36	0.35	36	0.35	0.00	3.69

Note: FeasIt represents feasible solution iteration; FeasTi represents feasible solution time; BestIt represents best solution iteration; BestTi represents best solution time.

(2) *Speed differences between fast and slow trains*

Since the speed differences between the fast and slow trains have a great impact on the difficulty in solving the cyclic train timetabling problem, we increase the speed range of the slow train by a 10 km/h step size from 240-260 km/h to 280-300 km/h, where the speed ranges of the fast trains are fixed to 280-300 km/h. Tables 13-15 show the lower bound and upper bound values, computation times and optimality gaps of two illustrative examples with 6 lines under different slow train speed ranges.

First, CPLEX still shows efficient and effective performance in obtaining the optimal solutions for the two illustrative examples with varying train speed range combinations. Second, it can be observed that Lagrangian relaxation cannot even obtain feasible solutions for illustrative example 1 with the speed ranges of slow trains be equal to 250-270 km/h and 260-280 km/h, showing the unstable performance of Lagrangian relaxation in achieving primal feasibility of the cyclic train timetabling problem. In addition, the symmetry issues in cyclic train timetabling problem are more significant due to the large departure time windows of the trains, which lead to the poor performance of priority rule-based heuristic in Lagrangian relaxation. By contrast, ADMM obtains good feasible solutions with the original optimality gaps are less than or equal to 1.5% for all of the test cases which further prove the advantages of symmetry breaking mechanism in ADMM. Moreover, both the best optimality gaps of Lagrangian relaxation and ADMM for most of the cases are slightly improved compared to the corresponding original optimality gaps. Third, the solving times of CPLEX decrease gradually as the speed differences between fast and slows trains are reduced, while there no obvious trend that smaller speed differences between fast and slow trains can result in smaller optimality gaps for Lagrangian relaxation and ADMM except for the solution results of ADMM in solving the illustrative example 1. Fourth, CPLEX spends fewer computation times in obtaining the feasible solutions than ADMM for most of the cases, while ADMM requires fewer computation times to find the best solutions than CPLEX except when the speed ranges of slow trains in illustrative example 2 are equal to 240-260 km/h, 270-290 km/h, and 280-300 km/h, where ADMM takes 649, 230, and 967 iterations to find the corresponding best solutions.

Table 13
Lower bound and upper bound values (in minutes) of two illustrative example with 6 lines under different slow train speed ranges

Slow train speed range (km/h)		LR-M3-C++		ADMM-M4-C++		PESP-CPLEX	
		Lower bound	Upper bound	Lower bound	Upper bound	Lower bound	Upper bound
Illustrative example 1	240-260	984.95	1029	1007	1019	1019	1019
	250-270	975.6	-	996.6	1003	1003	1003
	260-280	951.4	-	986.2	987	987	987
	270-290	949.4	1006	977	977	977	977
	280-300	925.8	989	968	968	968	968

Illustrative example 2	240-260	1040.85	1083	1062.85	1071	1066	1066
	250-270	1015.6	1069	1046.3	1062	1052	1052
	260-280	993.8	1055	1032.82	1038	1037	1037
	270-290	987.9	1023	1022	1023	1023	1023
	280-300	963.45	1010	1010	1010	1010	1010

Table 14
Computation times (in seconds) and optimality gaps of two illustrative example with 6 lines under different slow train speed ranges

Slow train speed range (km/h)		LR-M3-C++			ADMM-M4-C++			PESP-CPLEX		
		CPU time	GAP (%)	BestGAP (%)	CPU time	GAP (%)	BestGAP (%)	CPU time	GAP (%)	BestGAP (%)
Illustrative example 1	240-260	14.45	4.47	0.98	12.93	1.19	0	10.7	0	0
	250-270	13.28	-	-	12.13	0.64	0	6.1	0	0
	260-280	12.33	-	-	11.39	0.08	0	1.3	0	0
	270-290	12.27	5.96	2.97	11.18	0	0	1.3	0	0
	280-300	14.15	6.83	2.17	12.38	0	0	1	0	0
Illustrative example 2	240-260	13.01	4.05	1.59	10.95	0.77	0.47	5.2	0	0
	250-270	10.7	5.26	1.62	9.83	1.5	0.95	5.3	0	0
	260-280	10.33	6.16	1.74	9.71	0.5	0.1	1.1	0	0
	270-290	10.62	3.55	0	9.18	0.1	0	0.6	0	0
	280-300	11.05	4.83	0	10.05	0	0	0.8	0	0

Table 15
Feasible and best solution iterations as well as solution times (in seconds) of two illustrative with 6 lines under different slow train speed ranges

Slow train speed range (km/h)		LR-M3-C++				ADMM-M4-C++				PESP-CPLEX	
		FeasIt	FeasTi	BestIt	BestTi	FeasIt	FeasTi	BestIt	BestTi	FeasTi	BestTi
Illustrative example 1	240-260	6	0.09	6	0.09	9	0.12	214	2.77	0.00	3.1
	250-270	-	-	-	-	8	0.1	8	0.1	0.03	5.77
	260-280	-	-	-	-	1	0.01	1	0.01	0.00	0.12
	270-290	4	0.05	4	0.05	1	0.01	1	0.01	0.02	1.3
	280-300	4	0.06	25	0.35	1	0.01	1	0.01	0.00	1
Illustrative example 2	240-260	54	0.7	55	0.72	6	0.07	649	7.11	0.00	3.9
	250-270	3	0.03	333	3.56	319	3.14	334	3.28	0.00	4.37
	260-280	4	0.04	4	0.04	7	0.07	33	0.32	0.00	1.1
	270-290	409	4.34	941	9.99	229	2.1	230	2.11	0.00	0.6
	280-300	20	0.22	348	3.85	6	0.06	967	9.72	0.00	0.8

Note: FeasIt represents feasible solution iteration; FeasTi represents feasible solution time; BestIt represents best solution iteration; BestTi represents best solution time.

6.2 Real-life experiments based on the Beijing-Shanghai high-speed railway corridor

Beijing-Shanghai high-speed railway corridor in China is a double-track railway that consists of 23 stations and 22 segments, and it has a length of 1318 km. Fig. 7 shows the structure of the Beijing-Shanghai high-speed railway corridor where the Beijing South and Shanghai Hongqiao stations are two major terminal stations, and the numbers beside each station show the cumulative distance in kilometers starting from the Beijing South station. It can be seen from Fig. 7 that the segment from Cangzhou West station to Dezhou East station has the maximum length of 104 km, and the average distance between the two stations is 59.91 km.

In this study, we only consider the trains running from Beijing South station to Shanghai Hongqiao Station, and 12 scenarios are designed with the number of lines ranges from 10 to 34. The scenarios 1-4 have the number of lines of 10, 12, 14 and 17, respectively, and they adopt the line plan data in Zhang and Nie (2016). In addition, the speed ranges of trains in scenarios 1-4 are set to 310-330 km/h. For the scenarios 5-12, they are generated based on the actual train timetable data of the Beijing-Shanghai high-speed railway corridor where all trains run from Beijing South station to Shanghai Hongqiao station. Furthermore, the number of lines in scenarios 5-12 are equal to 20, 22, 24, 26, 28, 30, 32 and 34, respectively. In particular, scenario 12 consists of 36 trains including 2 fast trains and 34 slow trains. Moreover, the speed ranges of fast and slow trains in scenarios 5-12 are set to 310-330 km/h and 280-300 km/h, respectively. In addition, the parameters for headway requirements, minimum and maximum dwell times, and acceleration and deceleration times are also set to the values in Table 8. Besides, the cycle lengths T of scenarios 1-4, scenarios 5-9 and scenarios 10-12 are set to 160 min, 300 min and 360 min, respectively. Therefore, the values of the parameter H for the 12 scenarios are all set to 3. Furthermore, the lengths of T_m and T_e for scenarios 1-4, 5-9 and 10-12 are equal to 480 min and 960 min, 900 min and 1800 min, and 1080 min and 2160 min, respectively. The maximum number of iterations K for Lagrangian relaxation and ADMM are set to 1000 and the time limit of CPLEX is set to 10800 seconds for all of the test cases. Tables 16-19 list the detailed experimental results of those 12 scenarios.

For scenarios 1-4, it can be shown that both Lagrangian relaxation and ADMM can obtain quite good solution results, while ADMM achieves better lower bound and upper bound values than Lagrangian relaxation, and thus ADMM has smaller original optimality gaps. However, when looking at the values of best optimality gaps in Table 17, it is found that the best optimality gaps of Lagrangian relaxation are much smaller than the corresponding original optimality gaps. Moreover, CPLEX obtains optimal solutions

for the first three test cases and closes to the optimal solution for the last test case, resulting in the best lower bound and upper bound values and thus smallest original and best optimality gaps. On the other hand, the computation times of CPLEX grow sharply when the number of lines increases gradually. By contrast, the computation times of Lagrangian relaxation and ADMM increase slowly and the advantages of them are more obvious for scenarios 3 and 4. Moreover, the number of iterations for both Lagrangian relaxation and ADMM to obtain the feasible and best solutions grows quickly as the number of lines increases, while most of the corresponding feasible and best solution times are much smaller than those of CPLEX due to the high efficiency of Lagrangian relaxation and ADMM in performing the computation tasks of each iteration. In addition, the computation times of CPLEX in obtaining the feasible and best solutions also increase rapidly, especially for scenario 4 where the best solution time is even close to the time limit of 10800 seconds.

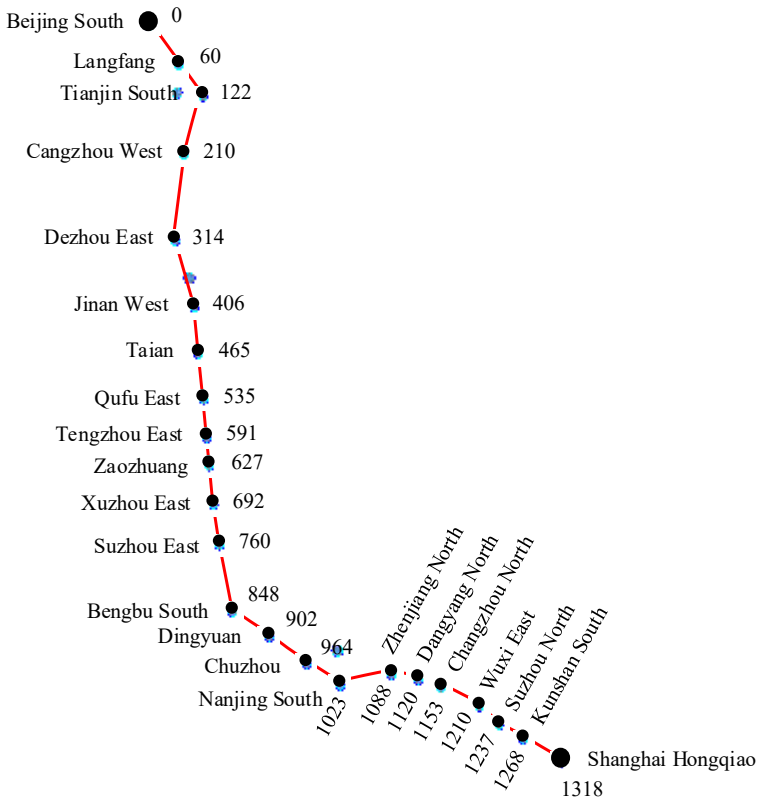


Fig. 7. Structure of the Beijing-Shanghai high-speed railway corridor

In addition, Fig. 8 depicts the convergence process of the best lower bound and upper bound values of Lagrangian relaxation and ADMM for scenario 4 including 17 lines and 18 trains with iteration number. At the initial stage, both the best lower bound values of Lagrangian relaxation and ADMM increase rapidly from relatively small values. In Table 19, it can be shown that the priority rule-based heuristic of Lagrangian relaxation takes 31 iterations to find a feasible solution, and the best upper bound solution is found after another 755 iterations. On the other hand, when the value of penalty parameter ρ of ADMM is increased gradually according to the method in Sec. 5.1, a good best upper bound which is equal to that of Lagrangian relaxation is obtained after 216 iterations. In addition, it can be also seen from Table 16 that the best lower bound value of ADMM is very close to that of CPLEX and it is slightly better than that of Lagrangian relaxation, which is an important factor for the small optimality gap of ADMM. Finally, the cyclic train diagram of scenario 4 obtained by ADMM is illustrated in Fig. 11, showing the feasibility of the cyclic train timetabling model based on extended time-space network in Sec. 4.2.2.

Table 16
Lower bound and upper bound values (in minutes) of the nine scenarios with a different number of lines

Scenario	# of lines (trains)	LR-M3-C++		ADMM-M4-C++		PESP-CPLEX	
		Lower bound	Upper bound	Lower bound	Upper bound	Lower bound	Upper bound
1	10 (11)	2392.6	2532	2532	2532	2532	2532
2	12 (13)	2993.99	3186	3184	3184	3184	3184
3	14 (15)	3502.74	3713	3713	3713	3713	3713
4	17 (18)	4218.5	4629	4554.6	4629	4559	4608
5	20 (22)	6148.84	7141	7098	7098	7093	7175
6	22 (24)	6628.65	7823	7766	7766	7761	7813
7	24 (26)	6968.35	-	8422	8452	8417	8463

8	26 (28)	7401.54	-	9084	9180	9079	9286
9	28 (30)	7730.93	-	9739.7	9994	9735	-
10	30 (32)	8154.22	-	10408	10409	10397	-
11	32 (34)	8376.38	-	11076	11095	11065	-
12	34 (36)	8565.79	-	11730	11786	11721	-

Table 17
Computation times (in seconds) and optimality gaps of the nine scenarios with a different number of lines

Scenario	# of lines (trains)	LR-M3-C++			ADMM-M4-C++			PESP-CPLEX		
		CPU time	GAP (%)	BestGAP (%)	CPU time	GAP (%)	BestGAP (%)	CPU time	GAP (%)	BestGAP (%)
1	10 (11)	89.69	5.83	0	61.4	0	0	13.1	0	0
2	12 (13)	107.95	6.41	0.06	82.1	0	0	152.5	0	0
3	14 (15)	120.72	6	0	94.53	0	0	637.34	0	0
4	17 (18)	138.89	9.73	1.54	122	1.63	1.54	10800	1.07	1.07
5	20 (22)	385.38	16.14	0.61	297.24	0	0	10800	1.16	1.08
6	22 (24)	415.88	18.02	0.73	322.86	0	0	10800	0.67	0.61
7	24 (26)	446.63	-	-	480.5	0.36	0.36	10800	0.55	0.49
8	26 (28)	478.37	-	-	527.05	1.06	1.06	10800	2.28	2.22
9	28 (30)	482.28	-	-	549.44	2.61	2.61	10800	-	-
10	30 (32)	678.95	-	-	715.04	0.01	0.01	10800	-	-
11	32 (34)	695.37	-	-	774.29	0.17	0.17	10800	-	-
12	34 (36)	725.29	-	-	816.58	0.48	0.48	10800	-	-

Table 18
Computation times (in seconds) of lower bound and upper bound (ADMM) solutions of the nine scenarios with a different number of lines

Scenario	# of lines (trains)	LR-M3-C++			ADMM-M4-C++		
		Lower bound solution	Upper bound solution	Other steps	Lower bound solution	ADMM solution	Other steps
1	10 (11)	28.92	57.13	3.64	20.58	39.02	1.8
2	12 (13)	37.32	69.62	1.01	22.96	56.54	2.6
3	14 (15)	43.39	76.35	0.98	26.47	65.22	2.84
4	17 (18)	54.26	83.64	0.99	33.87	84.82	3.31
5	20 (22)	135.58	248	1.8	85.73	204.84	6.67
6	22 (24)	149.22	264.89	1.77	92.59	222.56	7.71
7	24 (26)	161.48	283.35	1.8	158.59	313.96	7.95
8	26 (28)	178.15	298.41	1.81	170.07	337.43	19.55
9	28 (30)	184.2	296.36	1.72	178.21	362.51	8.72
10	30 (32)	247.66	429.04	2.25	231.06	472.89	11.09
11	32 (34)	257.9	435.29	2.18	249.33	513.29	11.67
12	34 (36)	273.5	449.6	2.19	261.77	541.96	12.85

Table 19
Feasible and best solution iterations as well as solution times (in seconds) of the nine scenarios with a different number of lines

Scenario	# of lines (trains)	LR-M3-C++				ADMM-M4-C++				PESP-CPLEX	
		FeasIt	FeasTi	BestIt	BestTi	FeasIt	FeasTi	BestIt	BestTi	FeasTi	BestTi
1	10 (11)	2	0.18	10	0.9	1	0.06	1	0.06	0.02	13.1
2	12 (13)	14	1.51	236	25.48	4	0.33	70	5.75	7.97	152.5
3	14 (15)	22	2.66	883	106.6	16	1.51	633	59.84	7.88	607.74
4	17 (18)	31	4.31	786	109.17	216	26.35	216	26.35	2638.03	9084.97
5	20 (22)	16	6.17	453	174.58	1	0.3	101	30.02	2228.26	6827.09
6	22 (24)	2	0.83	895	372.21	18	5.81	46	14.85	6513.36	8209.78
7	24 (26)	-	-	-	-	54	25.95	54	25.95	9178.57	10387.92
8	26 (28)	-	-	-	-	419	220.83	685	361.03	10163.78	10670.38
9	28 (30)	-	-	-	-	274	150.55	274	150.55	-	-
10	30 (32)	-	-	-	-	19	13.59	19	13.59	-	-
11	32 (34)	-	-	-	-	46	35.62	46	35.62	-	-
12	34 (36)	-	-	-	-	169	138	169	138	-	-

Note: FeasIt represents feasible solution iteration; FeasTi represents feasible solution time; BestIt represents best solution iteration; BestTi represents best solution time.

For the scenarios 5-9, they show more difficulty in obtaining good feasible solutions due to a larger number of trains and cycle length T. The original optimality gaps of Lagrangian relaxation for the scenarios 5 and 6 increase to about 16% and Lagrangian relaxation even cannot generate feasible solutions for scenarios 7- 9. Furthermore, the best optimality gaps of Lagrangian relaxation decrease to about 0.6% compared to the corresponding original optimality gaps, which further shows that the low quality of lower

bound solutions of Lagrangian relaxation. Moreover, CPLEX reaches the time limit of 10800 seconds for all of the five scenarios and the times for CPLEX to find the best upper solutions also increase rapidly. In addition, CPLEX cannot obtain a feasible solution for scenario 9 with 28 lines. By contrast, ADMM not only achieves smaller original optimality gaps than CPLEX, but it also obtains a feasible solution with a 2.61% original optimality gap for the last and most difficult test case. Furthermore, the computation times of Lagrangian relaxation in calculating the lower bound solutions are slightly larger than those of ADMM, while ADMM requires more times to update the ADMM solutions for scenarios 7-9 than Lagrangian relaxation for calculating the upper bound solutions. Besides, both Lagrangian relaxation and ADMM are more likely to take more iterations to obtain the best solutions compared to those of scenarios 1-4. The convergence curves of Lagrangian relaxation and ADMM for the scenario 9 with 30 trains are depicted in Fig. 9, which show a similar convergence pattern with that of scenario 4 in Fig. 8. In particular, the priority rule-based heuristic of Lagrangian relaxation fails to find a feasible solution for scenarios 9, and we only illustrate how the best lower bound values of Lagrangian relaxation evolves with iteration number. Besides, Fig. 12 visualizes the cyclic train diagram of scenario 9 obtained by ADMM, where the trains turn out to be slightly congested than that in Fig. 11.

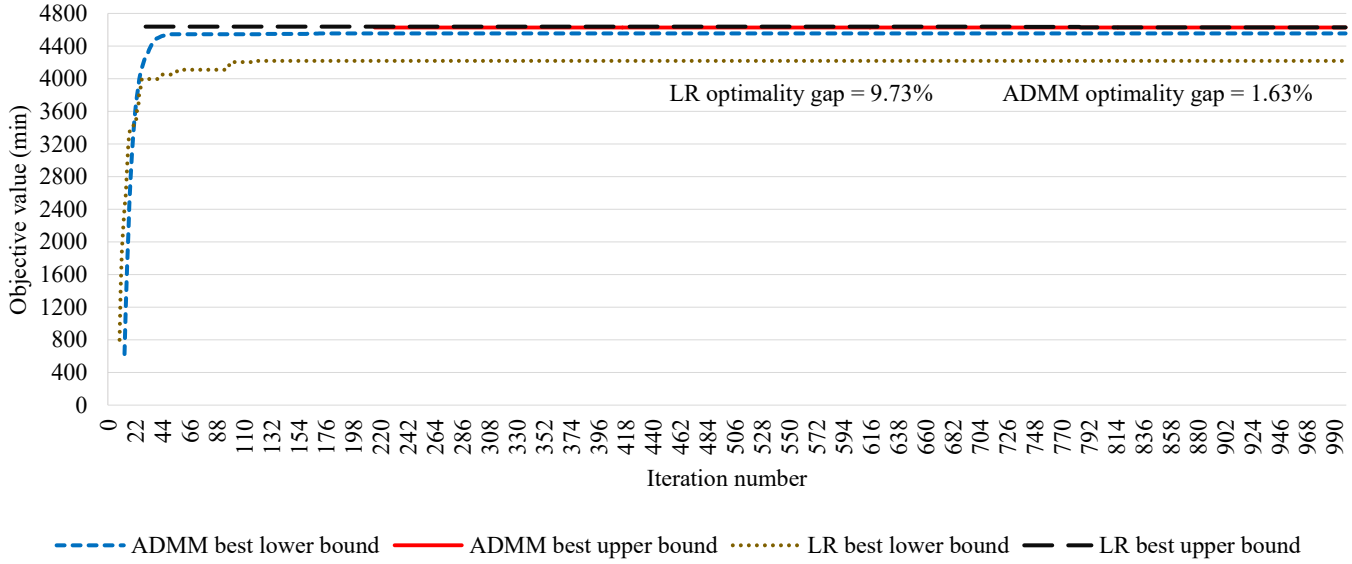


Fig. 8. Best lower bound and upper bound values of ADMM and LR for the scenario 4 with iteration number (17 lines, 18 trains)

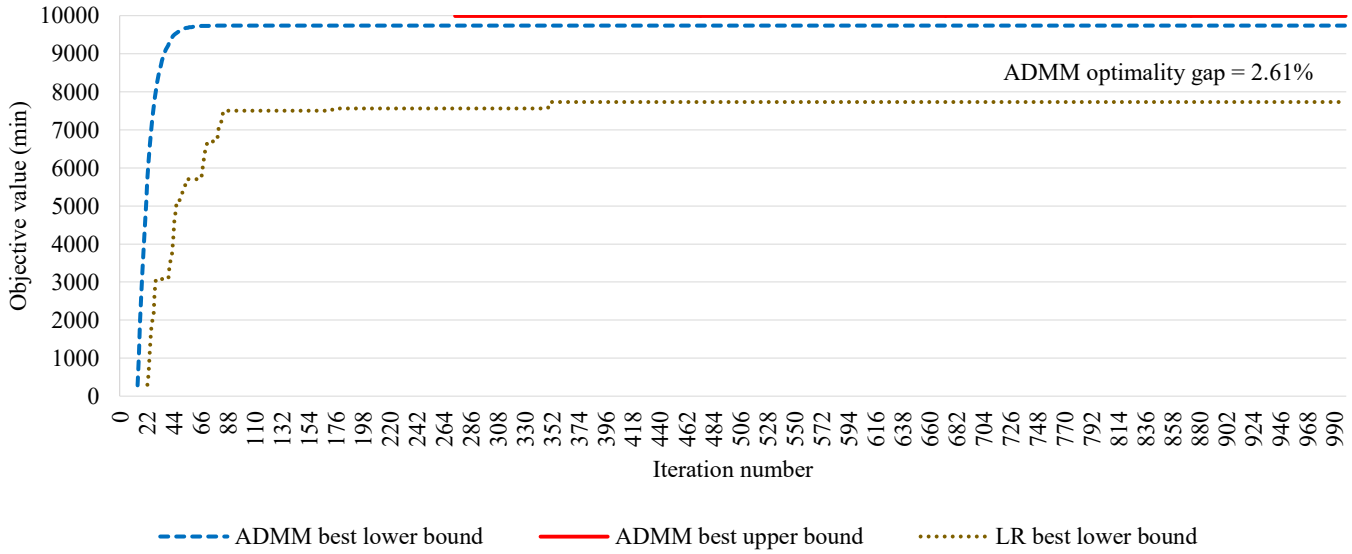


Fig. 9. Best lower bound and upper bound values of ADMM and LR for the scenario 9 with iteration number (28 lines, 30 trains)

For scenarios 10-12, they have the largest cycle length T and more trains to be scheduled. As a result, both Lagrangian relaxation and CPLEX cannot find a feasible solution for all of the three scenarios. By contrast, ADMM can provide the best lower bound and upper bound values for scenarios 10-12. Therefore, it can be concluded that the subgradient method and the priority rule-based heuristic in Lagrangian relaxation as well as the embedded branch-and-cut algorithm in CPLEX do not have good scalability on

larger scale problems, while ADMM still shows robust and good performance. Specifically, ADMM mainly has the advantages of (1) the symmetry breaking mechanism, (2) simultaneously improvement on the primal and dual feasibility (Boyd et al., 2011; Yao et al., 2019), (3) a special type of variable splitting technique to handle the periodicity constraints and (4) the efficient time-dependent least cost path algorithm with duplicated time-space variables.

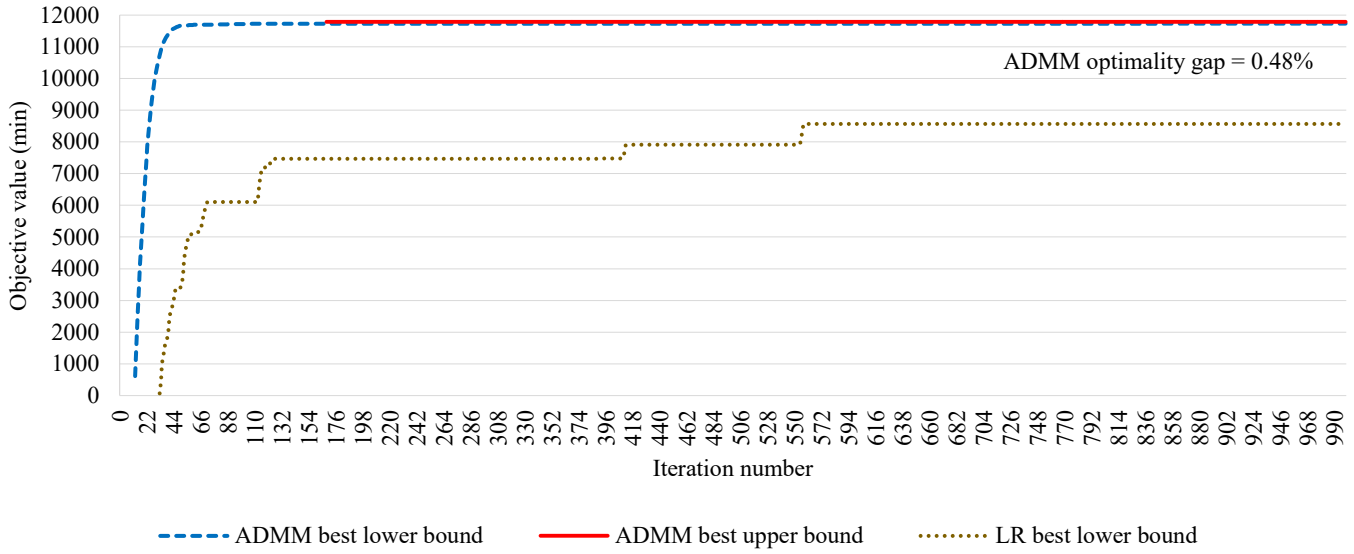


Fig. 10. Best lower bound and upper bound values of ADMM and LR for the scenario 12 with iteration number (34 lines, 36 trains)

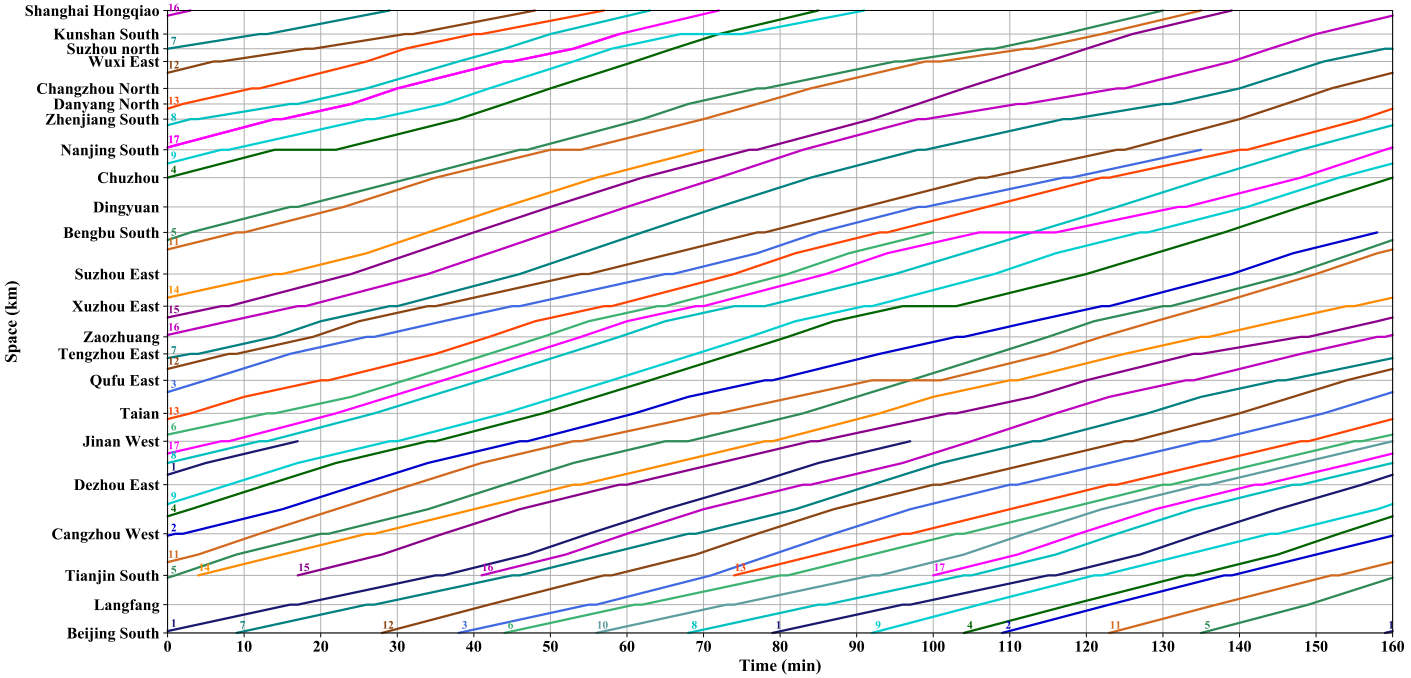


Fig. 11. Cyclic train diagram for scenario 4 obtained by ADMM with 17 lines and 18 trains (the numbers above the lines are the line numbers)

Meanwhile, it can be observed in Figures 8-10 that both the best lower bound values of Lagrangian relaxation and ADMM converge quite rapidly to the final values, and the best lower bound values of Lagrangian relaxation in Meng and Zhou (2014) also shows a similar convergence pattern. Therefore, it can be concluded that the usage of the subgradient method of Lagrangian relaxation in Eq. (17) and a similar method of ADMM in Eq. (21) for updating the values of Lagrangian multipliers have led to the rapid convergence of best lower bound values. Furthermore, it can be shown from Table 19 that the best upper bound solutions of ADMM for some scenarios turn out to be the feasible solutions found for the first time, while ADMM still needs a lot of iterations to improve the initial feasible solutions for some other scenarios. Therefore, ADMM could be stuck at the local optimum solutions or stationary points, and we believe it is necessary to perform intelligent branching operations within a branch-and-bound solution framework to further reduce the optimality gap of ADMM (Fisher, 1981; Yang and Zhou, 2014). In addition, Fig. 13 illustrates the cyclic train

diagram of scenario 12 obtained by ADMM, where there are almost no overtaking operations compared to the cyclic train diagram in Fig. 12.

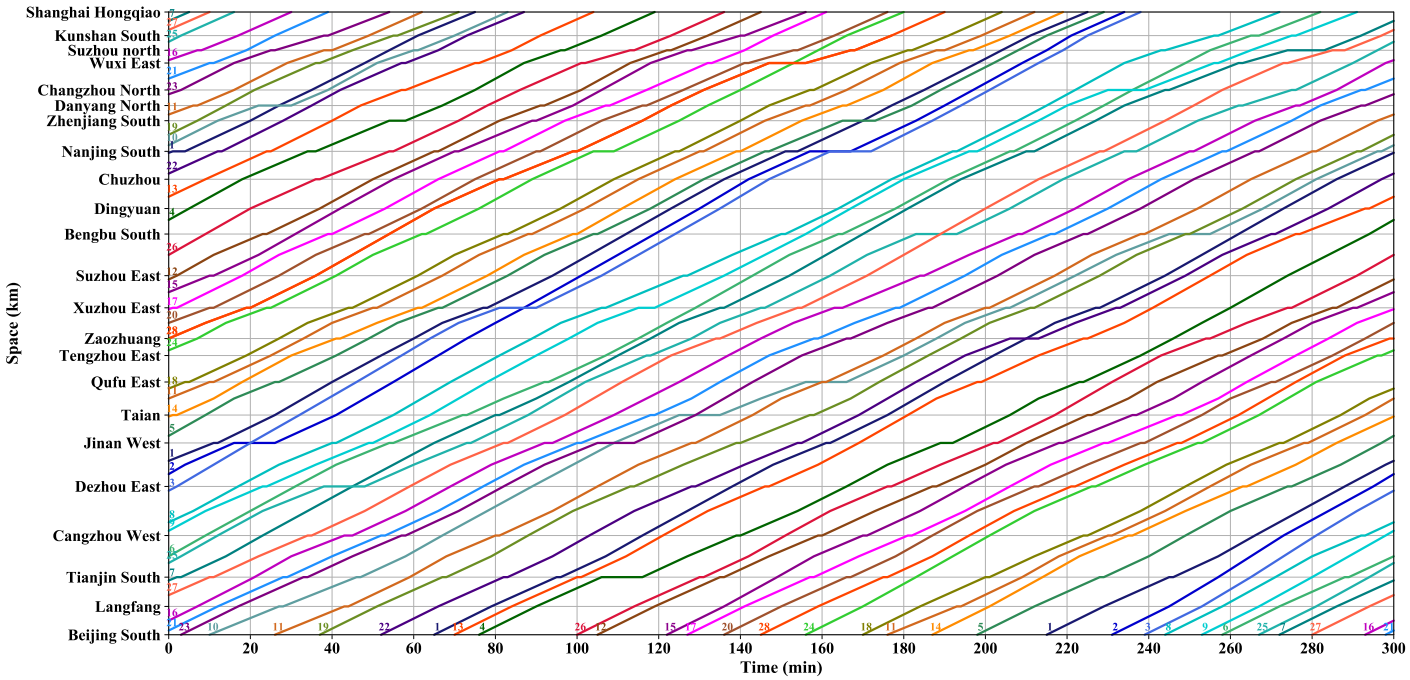


Fig. 12. Cyclic train diagram for scenario 9 obtained by ADMM with 28 lines and 30 trains (the numbers above the lines are the line numbers)

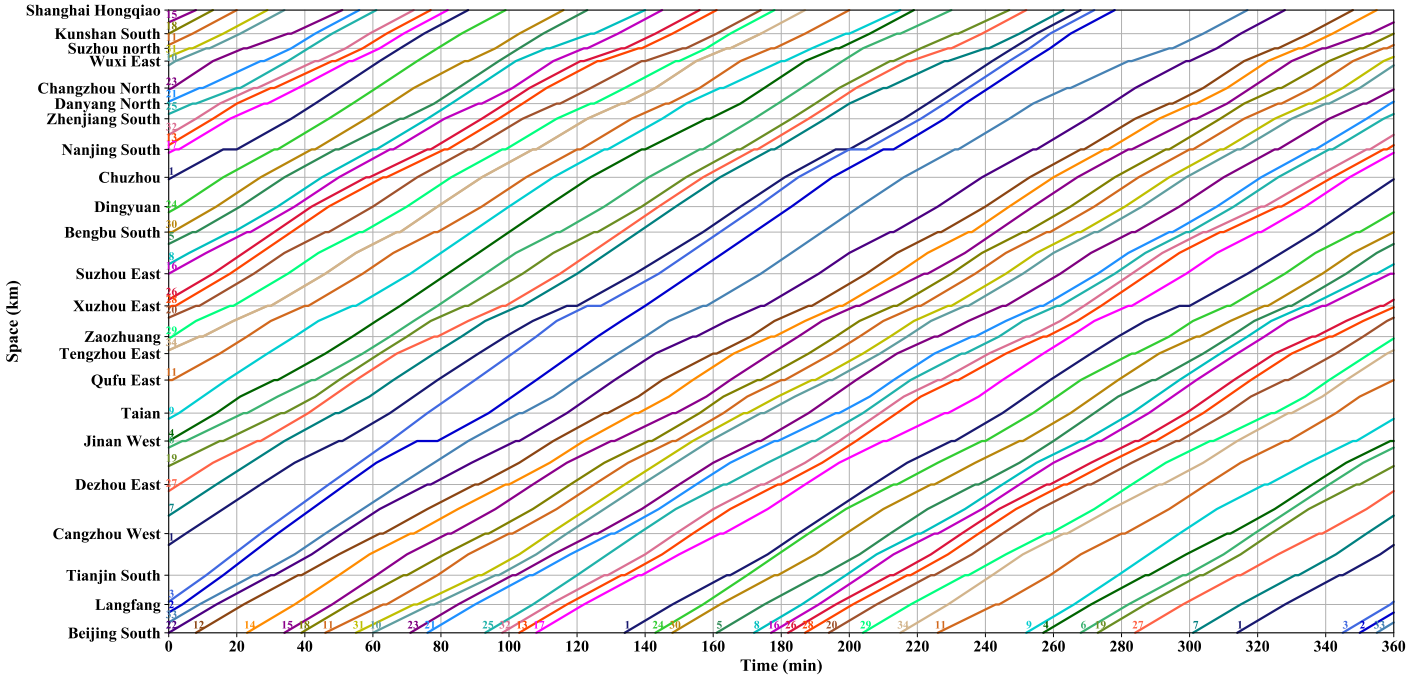


Fig. 13. Cyclic train diagram for scenario 12 obtained by ADMM with 34 lines and 36 trains (the numbers above the lines are the line numbers)

7. Conclusion

In this study, a new cyclic train timetabling model based on the extended time-space network is proposed for the double-track railway corridor at the macroscopic level. By considering the special specifications for the cyclic train timetabling problem that has been well described in the PESP based integer programming model, the concepts of master and extended schedules are introduced into the general time-space network model for the non-cyclic train timetabling problem. In particular, the master schedule allows the trains to go through the cyclic boundary without using modulo variables in the PESP model, and the extended schedule with duplicated time-space variables can guarantee a conflict-free cyclic train timetable. As a result, the new cyclic train timetabling model is formulated with additional master schedule coupling and extended schedule duplication constraints. Moreover, the side

track capacity constraints are dualized by using the two dual decomposition approaches including Lagrangian relaxation and ADMM. The quadratic term in ADMM can be nicely linearized according to the property of binary time-space arc selection variables, where the potential influences of all the other trains except the current train can be reflected directly with the rolling update scheme of time-space arc usage costs in ADMM. In addition, each train-specific sub-problem in Lagrangian relaxation and ADMM can be solved efficiently by adopting a modified version of the forward dynamic programming algorithm with duplicated time-space variables. Besides, a set of illustrative and real-life experiments are performed to test the efficiency and effectiveness of the proposed new cyclic train timetabling model and the solution algorithms, where the PESP model is solved by the CPLEX solver to serve as the benchmarks. The test results show that the ADMM-based solution method outperforms the Lagrangian relaxation-based solution method, where the priority rule in Lagrangian relaxation even cannot obtain a feasible solution for some difficult cases. Furthermore, the symmetry breaking mechanism together with the penalty parameter value lifting strategy in ADMM can better achieve the primal and dual feasibility of the problem (Boyd et al., 2011; Yao et al., 2019), where the optimality gaps of ADMM for most of the test cases are relatively small. In particular, ADMM can obtain a near optimal solution for the most difficult case, while CPLEX cannot generate a feasible solution with the time limit of 3 hours.

The future work of this study can be extended in several interesting directions. First, it is assumed that the time-space trajectories of the trains belonging to the same line are evenly distributed, while slightly relaxation on this particular requirement can avoid unnecessary train dwell times which is beneficial to the capacity utilization of the cyclic train timetable (Zhang and Nie, 2016; Yan et al. 2019). Second, most of the previous research focuses on the cyclic train timetabling problem at the macroscopic level and very limited number of research addresses the simultaneous routing and scheduling of trains in the cyclic train timetable (Petering et al., 2015), where the proposed method in this study can be further extended to handle this issue (Meng and Zhou, 2014). Third, an additional "state" dimension can be introduced into the cyclic train timetabling model based on the extended time-space network, where the energy-efficient train movement (Wang et al., 2016; Zhou et al., 2017), time-varying passenger demand (Lu et al., 2018; Shang et al., 2018, 2019; Meng and Zhou, 2019) and integration of locomotive assignment (Xu et al., 2018) can be also considered in the cyclic train timetable. Fourth, the optimality gap of ADMM can be further reduced by incorporating it into a branch-and-bound solution framework (Fisher, 1981; Yang and Zhou, 2014) and the values of Lagrangian multipliers can be updated with more effective methods, such as the cutting plane method (Kallehauge et al., 2001; Kallehauge, 2008; Lubin et al., 2013) where the values of Lagrangian multipliers may be not unique at each iteration. Fifth, the robust cyclic train timetabling problem can be studied for a robust master schedule under day-dependent train travel times (Xing and Zhou, 2013), or for a robust cyclic train timetable with scenario-based stochastic train travel times (Yang and Zhou, 2014).

Acknowledgments

This work is supported by National Key Research and Development Program of China (No. 2017YFB1200700-1), National Natural Science Foundation of China (No. 61603317, No. U1834209, No. 71871188) and the Open Fund Project of Chongqing Key Laboratory of Traffic & Transportation (2018TE01). The first author is deeply grateful for the financial support from the China Scholarship Council (201707000080). The last author is partially funded by National Science Foundation–United States under NSF Grant No. CMMI 1663657. "Real-time Management of Large Fleets of Self-Driving Vehicles Using Virtual Cyber Tracks".

Appendix A. Headway requirements for the double-track high-speed railway corridor in China

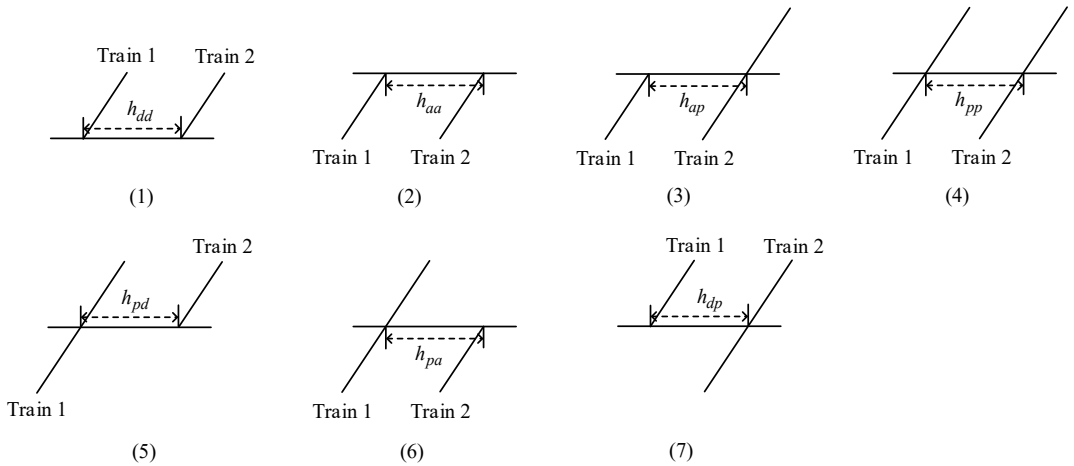


Fig. A1. Seven types of headway requirements for the high-speed railway corridor in China

Fig. A1 shows seven types of headway requirements between the arrival, departure and passing times of two trains (i.e., train 1 and train 2) in the same station. Specifically, the headway between two departure trains (h_{dd}), headway between two arrival trains (h_{aa}), headway between one arrival train and one passing train (h_{ap}), headway between two passing trains (h_{pp}), headway between

one passing trains and one departure train (h_{pd}), headway between one passing train and one arrival train (h_{pa}) and headway between one departure train and one passing train (h_{dp}).

Appendix B. Comparison of model M2 with the modified PESP-based cyclic train timetabling model in Zhang and Nie (2016)

The PESP-based model in Zhang and Nie (2016) is based on the periodic event-activity network. Specifically, the set of events ε is indexed by i, j, i' and j', which consists of the arrival events ε_{arr} and the departure events ε_{dep}. The set of activities $\mathcal{A}$ is indexed by (i,j) that includes the running activities $\mathcal{A}_{\text{run}}$, dwelling activities $\mathcal{A}_{\text{dwell}}$, passing activities $\mathcal{A}_{\text{pass}}$, safety activities $\mathcal{A}_{\text{safe}}$ and regular activities $\mathcal{A}_{\text{regular}}$. The complete modified PESP-based model in Zhang and Nie (2016) with the objective of minimizing the total train journey times and fixed cycle length T is listed as follows.

Objective function:

$$\min \sum_{(i,j)\in\mathcal{A}_{\text{run}}\cup\mathcal{A}_{\text{dwell}}} x_{ij}. \tag{B1}$$

Subject to:

$$x_{ij} = \pi_j - \pi_i + z_{ij}T, \quad \forall (i,j) \in \mathcal{A}_{\text{run}} \cup \mathcal{A}_{\text{dwell}} \cup \mathcal{A}_{\text{pass}} \cup \mathcal{A}_{\text{safe}} \cup \mathcal{A}_{\text{regular}} \tag{B2}$$

$$x_{ij} = 0, \quad \forall (i,j) \in \mathcal{A}_{\text{pass}} \tag{B3}$$

$$l_{ij} \le x_{ij} \le u_{ij}, \quad \forall (i,j) \in \mathcal{A}_{\text{run}} \cup \mathcal{A}_{\text{dwell}} \tag{B4}$$

$$h_{ij} \le x_{ij} \le T - h_{ij}, \quad \forall (i,j) \in \mathcal{A}_{\text{safe}} \tag{B5}$$

$$\frac{T}{f_{ij}} - \frac{K}{f_{ij}} \le x_{ij} \le \frac{T}{f_{ij}} + \frac{K}{f_{ij}}, \quad \forall (i,j) \in \mathcal{A}_{\text{regular}} \tag{B6}$$

$$z_{ij} = 0, \quad \forall (i,j) \in \mathcal{A}_{\text{regular}^*} \tag{B7}$$

$$z_{ij} \in \{0,1\}, \quad \forall (i,j) \in \mathcal{A}_{\text{run}} \cup \mathcal{A}_{\text{dwell}} \cup \mathcal{A}_{\text{safe}} \cup \mathcal{A}_{\text{regular}} \tag{B8}$$

$$0 \le \pi_i \le T - 1, \quad \forall i \in \varepsilon \tag{B9}$$

$$0 \le x_{ij} \le T - 1, \quad \forall (i,j) \in \mathcal{A} \tag{B10}$$

$$z_{ij} + z_{i'j'} + z_{ii'} + z_{jj'} = 2 \times \left(w_{ii'jj'} + v_{ii'jj'}\right), \quad \forall (i,j),(i',j') \in \mathcal{A}_{\text{run}}, (i,i'),(j,j') \in \mathcal{A}_{\text{safe}} \cup \mathcal{A}_{\text{regular}} \tag{B11}$$

$$-x_{ii'} + \left[\left(h_{ij} + h_{i'j'}\right) - l_{ii'}^{without}\right] \times s_{ii'} \le -l_{ii'}^{without}, \ \forall (i,i') \in \mathcal{A}_{\text{dwell}^*}, (i,j),(i',j') \in \mathcal{A}_{\text{safe}} \tag{B12}$$

$$x_{ii'} + \left(u_{ii'}^{without} - u_{ii'}\right) \times s_{ii'} \le u_{ii'}^{without}, \quad \forall (i,i') \in \mathcal{A}_{\text{dwell}^*} \tag{B13}$$

$$\left(z_{ij} + z_{i'j'} + z_{ii'}\right) - 2 \times q_{ii'jj'} - s_{ii'jj'} = 0, \quad \forall (i,j),(i',j') \in \mathcal{A}_{\text{safe}}, (i,i') \in \mathcal{A}_{\text{dwell}^*}, (j,j') \in \mathcal{A}_{\text{pass}} \tag{B14}$$

$$s_{ii'} = \sum_{(j,j')\in\mathcal{A}_{\text{pass}}} s_{ii'jj'}, \quad \forall (i,j),(i',j') \in \mathcal{A}_{\text{safe}}, (i,i') \in \mathcal{A}_{\text{dwell}^*} \tag{B15}$$

The variable π_i denotes the event time, and z_{ij} is the binary modulo variable. Moreover, the variable x_{ij} represents the duration of activity $(i,j) \in \mathcal{A}$. The parameters u_{ij} and l_{ij} restrict the range of x_{ij} for the running and dwelling activities, and the parameter h_{ij} specifies the headway value for the safe activity. In addition, the frequency of a regular activity is denoted by f_{ij} and the non-negative parameter K is introduced for the relaxation of regularity requirements. Readers can refer to Zhang and Nie (2016) for detailed definitions of the variables and parameters. The modified objective of the model in Eq. (B1) is to minimize the total train journey times $\sum_{(i,j)\in\mathcal{A}_{\text{run}}\cup\mathcal{A}_{\text{dwell}}} x_{ij}$, and mainly four categories of constraints are defined from constraints (B2) to (B15). In particular, constraint (B9) restricts the event times to the range $[0, T)$, and the safety constraint (B5) simultaneously enforces two safety constraints $\pi_j - \pi_i \ge h_{ij}$ and $\pi_i - \pi_j + T \ge h_{ij}$ or $\pi_i - \pi_j \ge h_{ij}$ and $\pi_j - \pi_i + T \ge h_{ij}$ corresponding to before and after precedence relationships for each pair of incompatible events i and j. The detailed comparison of model **M2** with the modified PESP-based model is summarized in Table B1. It can be concluded from Table B1 that model **M2** can handle most of the practical constraints in the modified PESP-based model of Zhang and Nie (2016) except the relaxation of regulatory requirement, where the flexible overtaking in the stations are only possible when regularity is not strictly enforced.

Table B1

Comparison of model **M2** with the modified PESP-based model in Zhang and Nie (2016)

Modified PESP-based model in Zhang and Nie (2016)	Model **M2** in this study
Variable trip times and dwell times, constraint (B4)	By splitting each intermediate station into two dummy stations and constructing the time-space arcs from (i,τ) to (i',τ')
Safety headway requirements, constraint (B5)	Track capacity constraint (16)
Prevention of illegal overtaking in the sections, constraint (B11)	By inserting a dummy station in the middle of the sections with long distance and trains can only pass through the corresponding dummy stations with the minimum headway h_{pp}
Relaxation of regularity requirement, constraint (B6) and flexible overtaking in the stations, constraints (B12)-(B15)	Not considered in model **M2** and trains belonging to the same line are evenly distributed for simplicity

Appendix C. An illustrative example of ADMM with three blocks

Based on the example in Boyd et al. (2011) that applies ADMM to solve an optimization problem with two blocks, we further extend the example for solving a slightly more complicated optimization problem in Eqs. (C1) and (C2) with three blocks x, y and z, where A, B and D are the coefficient matrices and c is the constant term on the right-hand side.

$$\min f(x) + g(y) + h(z) \quad \text{(C1)}$$
$$Ax + By + Dz = c \quad \text{(C2)}$$

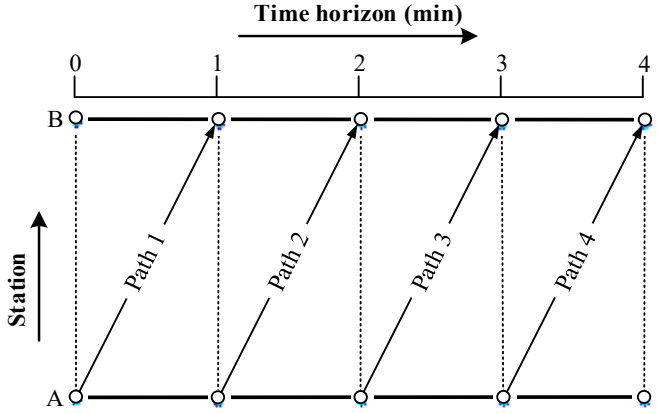
The equality constraint (C2) is dualized into the objective function with one linear term and one quadratic term by introducing the Lagrangian multipliers λ and penalty parameter ρ, and the resulting unconstrained optimization problem $L_\rho(x, y, z, \lambda)$ is denoted in Eq. (C3). Eqs. (C4)-(C7) illustrate the rolling update scheme for the three blocks x, y and z and the Lagrangian multipliers λ, where k represents the iteration number. Note that the update sequence of the three blocks is fixed in this example, and the value of penalty parameter ρ can be lifted gradually to obtain good feasible solutions.

$$L_\rho(x, y, z, \lambda) = f(x) + g(y) + h(z) + \lambda^T(Ax + By + Dz - c) + (\rho/2)\|Ax + By + Dz - c\|_2^2 \quad \text{(C3)}$$
$$x^{k+1} := argmin_x\, L_\rho(x, y^k, z^k, \lambda^k) \quad \text{(C4)}$$
$$y^{k+1} := argmin_y\, L_\rho(x^{k+1}, y, z^k, \lambda^k) \quad \text{(C5)}$$
$$z^{k+1} := argmin_z\, L_\rho(x^{k+1}, y^{k+1}, z, \lambda^k) \quad \text{(C6)}$$
$$\lambda^{k+1} := \lambda^k + \rho(Ax^{k+1} + By^{k+1} + Dz^{k+1} - c) \quad \text{(C7)}$$

Appendix D. Illustration of models M3 and M4 with a hypothetic example

Fig. D1 shows a hypothetic example for the illustration of models **M3** and **M4**. A hypothetic railway network in Fig. D1 (a) consists of two stations A and B with one section between them. In addition, two identical lines l_1 and l_2 need to be planned from stations A to B and both of them have the frequency of 1. Two trains a_1 and a_2 can be generated in associated with lines l_1 and l_2, respectively. The earliest departure and preferred arrival times of those two trains are equal to 0 min and 1 min respectively, and the section running times of them are fixed to 1 min. Besides, the values of safety headway parameters h_{dd} and h_{aa} are set to 1 min. According to the above parameter settings, four candidate time-space paths can be generated for trains a_1 and a_2 in Fig. D1 (b). Note that the absolute deviation from the preferred arrival time for each train are also penalized in the objective function, so that time-space arcs can be associated with different costs. The goal is to minimize the sum of running times and the absolute deviation from the preferred arrival times of those two trains.

(a) Artificial railway network with two trains a_1 and a_2



(b) Four candidate time-space paths for trains a_1 and a_2

Fig. D1. A hypothetic example for the illustration of models **M3** and **M4**

Table D1
Optimal upper bound feasible solution for the hypothetic example

Iteration 0:
Trains a_1 chooses path 1 and time-space vertexes (A, 0) and (B, 1) are marked as infeasible.
Path 1: $1 + (1 - 1) = 1$

Iteration 1:
Trains a_2 chooses path 2.
Path 2: $1 + (2 - 1) = 2$

Objective function value: $1 + 2 = 3$

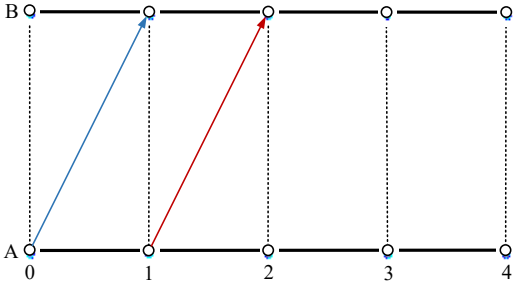


Table D1 gives the optimal upper bound feasible solution corresponding to the hypothetic example by the priority rule-based sequential method. Train a_1 is planned and it is assigned with the most favorable path 1 and the time-space vertexes (A, 0) and (B, 1) are marked as infeasible. Hence, train a_2 will choose path 2 instead with a cost of 2. Finally, the objective function value of the upper bound solution is equal to 3.

Table D2 lists the detailed steps of applying model **M3** to obtain the lower bound dual solutions. At the iteration 0, the step size α^0 and resource costs of eight time-space vertexes are initialized with 1 and 0, respectively. The cost of each path is calculated by summing three items, namely, arc travel time, absolute deviation from the preferred arrival time and the resource costs of the time-space vertexes. For instance, the cost of path 1 is equal to $1 + (1-1) + \lambda^0_{\mathrm{A},0} + \lambda^0_{\mathrm{B},1} = 1$, where the first item "1" is the arc travel time, the second item "(1-1)" is the absolute deviation from the preferred arrival time and $\lambda^0_{\mathrm{A},0} + \lambda^0_{\mathrm{B},1}$ is the sum of the resource costs of time vertexes (A, 0) and (B, 1). Due to path 1 has the least cost, both trains a_1 and a_2 choose path 1 and the objective function value of the dual solution Z_3^0 is equal to 2. After iteration 0, the step size α^1 and resource costs of all time-space vertexes at the iteration 1 are updated according to Eqs. (23) and (24), respectively. By following the above calculation procedure, it can be obtained that both trains a_1 and a_2 are assigned on path 1 and the objective function value Z_3^1 equals to 3. Note that the optimality gap reaches 0 at iteration 1, however, the corresponding dual solution is not feasible because the identical path 2 is not used. In addition, another iteration 2 is performed and the results show that the symmetry issue still exist and both trains a_1 and a_2 select path 2.

Table D2

Lower bound dual solution updating process for applying model **M3** on the hypothetic example

Iteration 0:

$\alpha^0 = 1$

$\lambda^0_{\mathrm{A},0} = 0, \lambda^0_{\mathrm{A},1} = 0, \lambda^0_{\mathrm{A},2} = 0, \lambda^0_{\mathrm{A},3} = 0, \lambda^0_{\mathrm{A},4} = 0$

$\lambda^0_{\mathrm{B},0} = 0, \lambda^0_{\mathrm{B},1} = 0, \lambda^0_{\mathrm{B},2} = 0, \lambda^0_{\mathrm{B},3} = 0, \lambda^0_{\mathrm{B},4} = 0$

Path 1: $1 + (1-1) + \lambda^0_{\mathrm{A},0} + \lambda^0_{\mathrm{B},1} = 1$

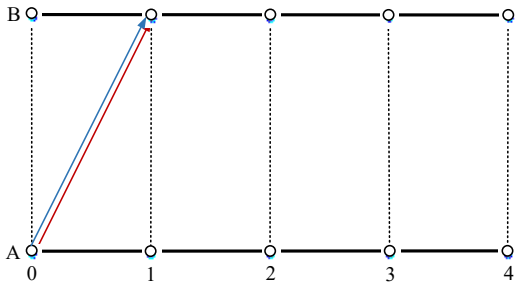
Path 2: $1 + (2-1) + \lambda^0_{\mathrm{A},1} + \lambda^0_{\mathrm{B},2} = 2$

Path 3: $1 + (3-1) + \lambda^0_{\mathrm{A},2} + \lambda^0_{\mathrm{B},3} = 3$

Path 4: $1 + (4-1) + \lambda^0_{\mathrm{A},3} + \lambda^0_{\mathrm{B},4} = 4$

Path choices: both trains a_1 and a_2 choose the path 1

$Z_3^0 = 1 + 1 = 2$



Iteration 1:

$\alpha^1 = 1/2$

$\lambda^1_{\mathrm{A},0} = 1/2, \lambda^1_{\mathrm{A},1} = 0, \lambda^1_{\mathrm{A},2} = 0, \lambda^1_{\mathrm{A},3} = 0, \lambda^1_{\mathrm{A},4} = 0$

$\lambda^1_{\mathrm{B},0} = 0, \lambda^1_{\mathrm{B},1} = 1/2, \lambda^1_{\mathrm{B},2} = 0, \lambda^1_{\mathrm{B},3} = 0, \lambda^1_{\mathrm{B},4} = 0$

Path 1: $1 + (1-1) + \lambda^1_{\mathrm{A},0} + \lambda^1_{\mathrm{B},1} = 2$

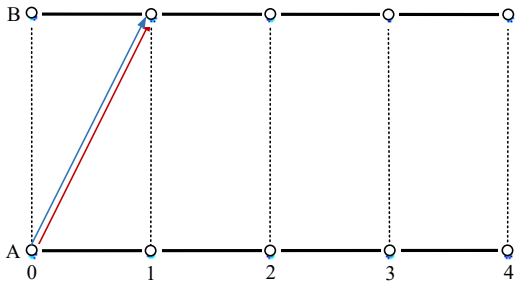
Path 2: $1 + (2-1) + \lambda^1_{\mathrm{A},1} + \lambda^1_{\mathrm{B},2} = 2$

Path 3: $1 + (3-1) + \lambda^1_{\mathrm{A},2} + \lambda^1_{\mathrm{B},3} = 3$

Path 4: $1 + (4-1) + \lambda^1_{\mathrm{A},3} + \lambda^1_{\mathrm{B},4} = 4$

Path choices: both trains a_1 and a_2 choose the path 1

$Z_3^1 = 2 + 2 - \lambda^1_{\mathrm{A},0} - \lambda^1_{\mathrm{B},1} = 3$



Iteration 2:

$\alpha^2 = 1/3$

$\lambda^2_{\mathrm{A},0} = 5/6, \lambda^2_{\mathrm{A},1} = 0, \lambda^2_{\mathrm{A},2} = 0, \lambda^2_{\mathrm{A},3} = 0, \lambda^2_{\mathrm{A},4} = 0$

$\lambda^2_{\mathrm{B},0} = 0, \lambda^2_{\mathrm{B},1} = 5/6, \lambda^2_{\mathrm{B},2} = 0, \lambda^2_{\mathrm{B},3} = 0, \lambda^2_{\mathrm{B},4} = 0$

Path 1: $1 + (1-1) + \lambda^2_{\mathrm{A},0} + \lambda^2_{\mathrm{B},1} = 2.67$

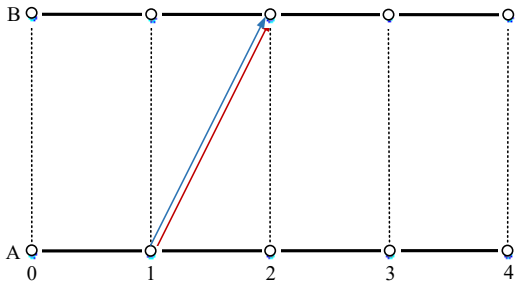
Path 2: $1 + (2-1) + \lambda^2_{\mathrm{A},1} + \lambda^2_{\mathrm{B},2} = 2$

Path 3: $1 + (3-1) + \lambda^2_{\mathrm{A},2} + \lambda^2_{\mathrm{B},3} = 3$

Path 4: $1 + (4-1) + \lambda^2_{\mathrm{A},3} + \lambda^2_{\mathrm{B},4} = 4$

Path choices: both trains a_1 and a_2 choose the path 2

$Z_3^2 = 2 + 2 - \lambda^2_{\mathrm{A},0} - \lambda^2_{\mathrm{B},1} = 2.33$



By contrast, the symmetry issue can be well handled by model **M4** with ADMM. Table D3 illustrates the calculation process of ADMM with two iterations. The penalty parameter ρ and resource costs of eight time-space vertexes are initialized with 2 and 0, respectively. At iteration 0, train a_1 is planned first and it can directly choose path 1 without any penalty due to no trains have been planned yet. However, at iteration 1, train a_2 will not select path 1 because the train a_1 has already occupied path 1 and a penalty term $(2/2)(2-1+2-1)$ is embedded into the cost of path 1. Hence, the cost of path 1 increases from 1 to 3 and path 2 which has

the cost of 2 turn out to be the least cost path for train a_2. It can be seen that the rolling update scheme together with the linearized penalty term in model **M4** with ADMM can nicely avoid the symmetry issue in model **M3** with Lagrangian relaxation, where Lagrangian relaxation allow trains to simultaneously find their least costs paths with the same resource costs.

Table D3

Lower bound dual solution updating process for applying model **M4** on the hypothetic example

Iteration 0:

$\rho = 2$

$\lambda^0_{A,0} = 0, \lambda^0_{A,1} = 0, \lambda^0_{A,2} = 0, \lambda^0_{A,3} = 0, \lambda^0_{A,4} = 0$

$\lambda^0_{B,0} = 0, \lambda^0_{B,1} = 0, \lambda^0_{B,2} = 0, \lambda^0_{B,3} = 0, \lambda^0_{B,4} = 0$

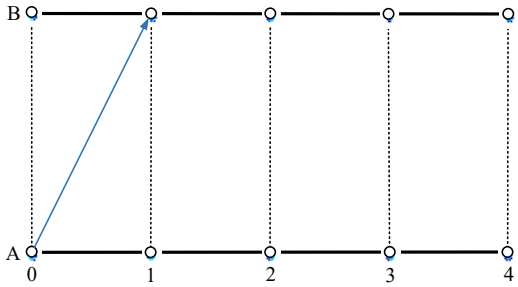
Path 1: $1 + (1-1) + \lambda^0_{A,0} + \lambda^0_{B,1} = 1$

Path 2: $1 + (2-1) + \lambda^0_{A,1} + \lambda^0_{B,2} = 2$

Path 3: $1 + (3-1) + \lambda^0_{A,2} + \lambda^0_{B,3} = 3$

Path 4: $1 + (4-1) + \lambda^0_{A,3} + \lambda^0_{B,4} = 4$

Path choices: trains a_1 chooses the path 1



Iteration 1:

$\rho = 2$

$\lambda^1_{A,0} = 0, \lambda^1_{A,1} = 0, \lambda^1_{A,2} = 0, \lambda^1_{A,3} = 0, \lambda^1_{A,4} = 0$

$\lambda^1_{B,0} = 0, \lambda^1_{B,1} = 0, \lambda^1_{B,2} = 0, \lambda^1_{B,3} = 0, \lambda^1_{B,4} = 0$

Path 1: $1 + (1-1) + \lambda^1_{A,0} + \lambda^1_{B,1} + (2/2)(2-1+2-1) = 3$

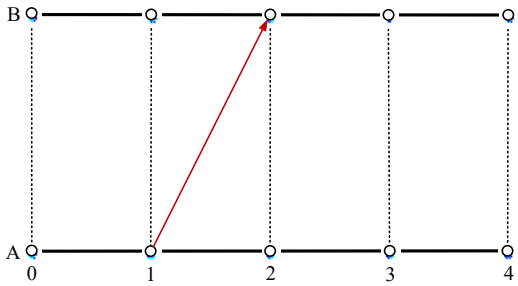
Path 2: $1 + (2-1) + \lambda^1_{A,1} + \lambda^1_{B,2} = 2$

Path 3: $1 + (3-1) + \lambda^1_{A,2} + \lambda^1_{B,3} = 3$

Path 4: $1 + (4-1) + \lambda^1_{A,3} + \lambda^1_{B,4} = 4$

Path choices: train a_2 chooses the path 2

$Z_4 = 1 + 2 = 3$



References

Assad, A. A., 1980. Models for rail transportation. *Transportation Research Part A: General*, 14(3), 205-220.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., Vance, P. H., 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3), 316-329.

Bertsekas, D. P., 1999. Nonlinear programming. Belmont: Athena scientific.

Bešinović, N., Goverde, R. M., Quaglietta, E., Roberti, R., 2016. An integrated micro–macro approach to robust railway timetabling. *Transportation Research Part B: Methodological*, 87, 14-32.

Brännlund, U., Lindberg, P. O., Nou, A., Nilsson, J. E., 1998. Railway timetabling using Lagrangian relaxation. *Transportation science*, 32(4), 358-369.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1), 1-122.

Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Linderoth, J., Luedtke, J., Oliveira, F., 2018. Combining Progressive Hedging with a Frank--Wolfe Method to Compute Lagrangian Dual Bounds in Stochastic Mixed-Integer Programming. *SIAM Journal on Optimization*, 28(2), 1312-1336.

Bożejko, W., Wodecki, M., 2018. On Cyclic Job Shop Scheduling Problem. In *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*. 000265-000270.

Burggraeve, S., Bull, S. H., Vansteenwegen, P., Lusby, R. M., 2017. Integrating robust timetabling in line plan optimization for railway systems. *Transportation Research Part C: Emerging Technologies*, 77, 134-160.

Cordeau, J. F., Toth, P., Vigo, D., 1998. A survey of optimization models for train routing and scheduling. *Transportation science*, 32(4), 380-404.

Chabini, I., 1998. Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. *Transportation Research Record: Journal of the Transportation Research Board*, 1645, 170–175.

Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and solving the train timetabling problem. *Operations research*, 50(5), 851-861.

Caprara, A., Monaci, M., Toth, P., Guida, P. L. 2006. A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete applied mathematics*, 154(5), 738-753.

Caprara, A., Kroon, L., Monaci, M., Peeters, M., Toth, P., 2007. Passenger railway optimization. *Handbooks in operations research and management science*, 14, 129-187.

Cacchiani, V., Caprara, A., Toth, P., 2008. A column generation approach to train timetabling on a corridor. *4OR*, 6(2), 125-142.

Cacchiani, V., Caprara, A., Toth, P., 2010. Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological*, 44(2), 215-231.

Cacchiani, V., Caprara, A., Fischetti, M., 2012. A Lagrangian heuristic for robustness, with an application to train timetabling. *Transportation Science*, 46(1), 124-133.
Caimi, G., Kroon, L., and Liebchen, C., 2017. Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, 6(4), 285-312.
Chinese National Bureau of Statistics, http://www.stats.gov.cn/.
Cordone, R., Redaelli, F., 2011. Optimizing the demand captured by a railway system with a regular timetable. *Transportation Research Part B: Methodological*, 45(2), 430-446.
Crainic, T. G., Laporte, G., 1997. Planning models for freight transportation. *European journal of operational research*, 97(3), 409-438.
Crainic, T. G., Fu, X., Gendreau, M., Rei, W., Wallace, S. W., 2011. Progressive hedging-based metaheuristics for stochastic network design. *Networks*, 58(2), 114-124.
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*.
Dorfman, M. J., Medanic, J., 2004. Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Research Part B: Methodological*, 38(1), 81-98.
Fisher, M. L., 1981. The Lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1), 1-18.
Fortin, M., Glowinski, R., 2000. Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems. Elsevier.
Gabay, D., Mercier, B., 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1), 17-40.
Gattermann, P., Großmann, P., Nachtigall, K., Schöbel, A., 2016. Integrating Passengers' Routes in Periodic Timetabling: A SAT approach. In *16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
Ghoseiri, K., Szidarovszky, F., Asgharpour, M. J., 2004. A multi-objective train scheduling model and solution. *Transportation research part B: Methodological*, 38(10), 927-952.
Glowinski, R., Marroco, A., 1975. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2), 41-76.
Goerigk, M., Schöbel, A., 2013. Improving the modulo simplex algorithm for large-scale periodic timetabling. *Computers & Operations Research*, 40(5), 1363-1370.
Großmann, P., Hölldobler, S., Manthey, N., Nachtigall, K., Opitz, J., Steinke, P., 2012. Solving periodic event scheduling problems with SAT. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, Berlin, Heidelberg.
Harrod, S., 2011. Modeling Network Transition Constraints with Hypergraphs. *Transportation Science*, 45(1), 81-97.
Harrod, S., 2012. A tutorial on fundamental model structures for railway timetable optimization. *Surveys in Operations Research and Management Science*, 17(2), 85-96.
Heydar, M., Petering, M. E., Bergmann, D. R., 2013. Mixed integer programming for minimizing the period of a cyclic railway timetable for a single track with two train types. *Computers & Industrial Engineering*, 66(1), 171-185.
Herrigel, S., Laumanns, M., Szabo, J., Weidmann, U., 2018. Periodic railway timetabling with sequential decomposition in the PESP model. *Journal of Rail Transport Planning & Management*.
Higgins, A., Kozan, E., Ferreira, L., 1996. Optimal scheduling of trains on a single line track. *Transportation research part B: Methodological*, 30(2), 147-161.
Huisman, D., Kroon, L. G., Lentink, R. M., and Vromans, M. J., 2005. Operations research in passenger railway transportation. *Statistica Neerlandica*, 59(4), 467-497.
Jiang, F., Cacchiani, V., Toth, P., 2017. Train timetabling by skip-stop planning in highly congested lines. *Transportation Research Part B: Methodological*, 104, 149-174.
Kallehauge, B., Larsen, J., Madsen, O. B., 2001. Lagrangean duality applied on vehicle routing with time windows-experimental results. Technical Report IMM-REP-2000-8, Informatics and Mathematical Modelling, Technical University of Denmark, DTU Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby.
Kallehauge, B., 2008. Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research*, 35(7), 2307-2330.
Kroon, L. G., Peeters, L. W., 2003. A variable trip time model for cyclic railway timetabling. *Transportation Science*, 37(2), 198-212.
Kroon, L., Huisman, D., Abbink, E., Fioole, P.-J., Fischetti, M., Maroti, G., Schrijver, A., Steenbeek, A., Ybema, R., 2009. The new Dutch timetable: the OR revolution. *Interfaces* 39 (1), 6–17.
Kroon, L. G., Peeters, L. W., Wagenaar, J. C., Zuidwijk, R. A., 2013. Flexible connections in pesp models for cyclic passenger railway timetabling. *Transportation Science*, 48(1), 136-154.

Kümmling, M., Großmann, P., Nachtigall, K., Opitz, J., Weiß, R., 2015. A state-of-the-art realization of cyclic railway timetable computation. *Public Transport*, 7(3), 281-293.
Larsson, T., Yuan, D., 2004. An augmented lagrangian algorithm for large scale multicommodity routing. *Computational Optimization and Applications*, 27(2), 187-215.
Liebchen, C., 2004. Symmetry for periodic railway timetables. *Electronic Notes in Theoretical Computer Science*, 92, 34-51.
Liebchen, C., 2006. Periodic Timetable Optimization in Public Transport. PhD thesis. TU Berlin.
Liebchen, C., Möhring, R. H., 2007. The modeling power of the periodic event scheduling problem: railway timetables—and beyond. In *Algorithmic methods for railway optimization*. Springer, Berlin, Heidelberg.
Liebchen, C., 2008. The first optimized railway timetable in practice. *Transportation Science*, 42(4), 420-435.
Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S., Prigge, A., 2010. Computing delay resistant railway timetables. *Computers & Operations Research*, 37(5), 857-868.
Lindner, T., 2000. Train schedule optimization in public rail transport. *Mathematics—Key Technology for the Future: Joint Projects Between Universities and Industry*, 703-716.
Li, P., Mirchandani, P., Zhou, X., 2015. Solving simultaneous route guidance and traffic signal optimization problem using space-phase-time hypernetwork. *Transportation Research Part B: Methodological*, 81, 103-130.
Liu, J., Zhou, X., 2016. Capacitated transit service network design with boundedly rational agents. *Transportation Research Part B: Methodological*, 93, 225-250.
Lu, K., Han, B., Zhou, X., 2018. Smart urban transit systems: from integrated framework to interdisciplinary perspective. *Urban Rail Transit*, 4(2), 49-67.
Lubin, M., Martin, K., Petra, C. G., Sandıkçı, B., 2013. On parallelizing dual decomposition in stochastic integer programming. *Operations Research Letters*, 41(3), 252-258.
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.
Lusby, R. M., Larsen, J., Ehrgott, M., Ryan, D., 2011. Railway track allocation: models and methods. *OR spectrum*, 33(4), 843-883
Macharis, C., Bontekoning, Y. M., 2004. Opportunities for OR in intermodal freight transport research: A review. *European Journal of operational research*, 153(2), 400-416.
Mees, A. I., 1991. Railway scheduling by network optimization. *Mathematical and Computer Modelling*, 15(1), 33-42.
Meng, L., Zhou, X., 2011. Robust single-track train dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach. *Transportation Research Part B: Methodological*, 45(7), 1080-1102.
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.
Meng, L., Zhou, X., 2019. An integrated train service plan optimization model with variable demand: A team-based scheduling approach with dual cost information in a layered network. *Transportation Research Part B: Methodological*, 125, 1-28.
Mathias, K., 2008. Models for Periodic Timetabling Thesis. Erasmus Research Institute of Management (ERIM).
Mor, A., Speranza, M. G., 2018. Vehicle routing problems over time: a survey. < https://www.researchgate.net/publication/ 328743816_Vehicle_routing_problems_over_time_a_survey > (last accessed, 11/19/2018).
Mu, S., Dessouky, M., 2011. Scheduling freight trains traveling on complex networks. *Transportation Research Part B: Methodological*, 45(7), 1103-1123.
Nachtigall, K., Voget, S., 1996. A genetic algorithm approach to periodic railway synchronization. *Computers & Operations Research*, 23(5), 453-463.
Nachtigall, K., Opitz, J., 2008. Solving periodic timetable optimisation problems by modulo simplex calculations. In *OASIcs-OpenAccess Series in Informatics* (Vol. 9). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
Niu, H., Zhou, X., Tian, X., 2018. Coordinating assignment and routing decisions in transit vehicle schedules: A variable-splitting Lagrangian decomposition approach for solution symmetry breaking. *Transportation Research Part B: Methodological*, 107, 70-101.
Oliveira, E., Smith, B. M., 2000. A job-shop scheduling model for the single-track railway scheduling problem, *School of Computing Research Report*, University of Leeds, England.
Odijk, M. A., 1996. A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research Part B: Methodological*, 30(6), 455-464.
Pallottino, S., Scutella, M. G., 1998. Shortest path algorithms in transportation models: classical and innovative aspects. In *Equilibrium and advanced transportation modelling*. Springer, Boston, MA. 245-281.
Peeters, L., 2003. Cyclic Railway Timetable Optimization TRAIL Thesis Series. Erasmus Research Institute of Management.
Petering, M. E., Heydar, M., Bergmann, D. R., 2015. Mixed-integer programming for railway capacity analysis and cyclic, combined train timetabling and platforming. *Transportation Science*, 50(3), 892-909.
Qi, J., Yang, L., Di, Z., Li, S., Yang, K., Gao, Y., 2018. Integrated optimization for train operation zone and stop plan with passenger distributions. *Transportation Research Part E: Logistics and Transportation Review*, 109, 151-173.
Robenek, T., Azadeh, S. S., Maknoon, Y., Bierlaire, M., 2017. Hybrid cyclicity: Combining the benefits of cyclic and non-cyclic timetables. *Transportation Research Part C: Emerging Technologies*, 75, 228-253.
Robenek, T., Azadeh, S. S., Maknoon, Y., de Lapparent, M., Bierlaire, M., 2018. Train timetable design under elastic passenger demand. *Transportation Research Part B: Methodological*, 111, 19-38.

Serafini, P., Ukovich, W., 1989. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4), 550-581.
Schöbel, A., 2007. Integer programming approaches for solving the delay management problem. In *Algorithmic methods for railway optimization*. Springer, Berlin, Heidelberg.
Schachtebeck, M., Schöbel, A., 2010. To wait or not to wait—and who goes first? Delay management with priority decisions. *Transportation Science*, 44(3), 307-321.
Shang, P., Li, R., Liu, Z., Yang, L., Wang, Y., 2018. Equity-oriented skip-stopping schedule optimization in an oversaturated urban rail transit network. *Transportation Research Part C: Emerging Technologies*, 89, 321-343.
Shang, P., Li, R., Guo, J., Xian, K., Zhou, X., 2019. Integrating Lagrangian and Eulerian observations for passenger flow state estimation in an urban rail transit network: A space-time-state hyper network-based assignment approach. *Transportation Research Part B: Methodological*, 121, 135-167.
Siebert, M., Goerigk, M., 2013. An experimental comparison of periodic timetabling models. *Computers & Operations Research*, 40(10), 2251-2259.
Socha, K., Knowles, J., Sampels, M., 2002. A max-min ant system for the university course timetabling problem. In *International Workshop on Ant Algorithms*. Springer, Berlin, Heidelberg. 1-13.
Sparing, D., Goverde, R. M., 2017. A cycle time optimization model for generating stable periodic railway timetables. *Transportation Research Part B: Methodological*, 98, 198-223.
Tong, L., Pan, Y., Shang, P., Guo, J., Xian, K., Zhou, X, 2019. Open-source public transportation mobility simulation engine DTALite-S: A discretized space–time network-based modeling framework for bridging multi-agent simulation and optimization. *Urban Rail Transit*, 1-16.
US Bureau of Transportation Statistics, United States Department of Transportation. https://www.bts.gov/.
Wang, Y., Zhang, M., Ma, J., Zhou, X., 2016. Survey on driverless train operation for urban rail transit systems. *Urban Rail Transit*, 2(3-4), 106-113.
Wei, Y., Avcı, C., Liu, J., Belezamo, B., Aydın, N., Li, P. T., Zhou, X., 2017. Dynamic programming-based multi-vehicle longitudinal trajectory optimization with simplified car following models. *Transportation Research Part B: Methodological*, 106, 102-129.
Xing, T., Zhou, X., 2013. Reformulation and solution algorithms for absolute and percentile robust shortest path problems. *IEEE Transactions on Intelligent Transportation Systems*, 14(2), 943-954.
Xu, X., Li, C. L., Xu, Z., 2018. Integrated train timetabling and locomotive assignment. *Transportation Research Part B: Methodological*, 117, 573-593.
Yao, Y., Zhu, X., Dong, H., Wu, S., Wu, H., Tong, L. C., Zhou, X., 2019. An ADMM-based Problem Decomposition Scheme for Vehicle Routing Problem with Time Windows. https://www.researchgate.net/publication/333560037_An_ADMM-based_Problem_Decomposition_Scheme_for_Vehicle_Routing_Problem_with_Time_Windows (last accessed, 06/14/2019).
Yang, L., Zhou, X., 2014. Constraint reformulation and a Lagrangian relaxation-based solution algorithm for a least expected time path problem. *Transportation Research Part B: Methodological*, 59, 22-44.
Yan, F., Goverde, R. M., 2017. Railway timetable optimization considering robustness and overtakings. In *Proceedings of the5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2017*, 291–296.
Yan, F., Bešinović, N., Goverde, R. M. P., 2019. Multi-objective periodic railway timetabling on dense heterogeneous railway corridors. *Transportation Research Part B: Methodological*, 125, 52-75.
Yin, Y., Li, D., Bešinović, N., Cao, Z., 2019. Hybrid demand-driven and cyclic timetabling considering rolling stock circulation for a bidirectional railway line. *Computer-Aided Civil and Infrastructure Engineering*, 34(2), 164-187.
Yue, Y., Wang, S., Zhou, L., Tong, L., Saat, M. R., 2016. Optimizing train stopping patterns and schedules for high-speed passenger rail corridors. *Transportation Research Part C: Emerging Technologies*, 63, 126-146.
Zhan, S., Kroon, L. G., Zhao, J., Peng, Q., 2016. A rolling horizon approach to the high speed train rescheduling problem in case of a partial segment blockage. *Transportation Research Part E: Logistics and Transportation Review*, 95, 32-61.
Zhang, X., Nie, L., 2016. Integrating capacity analysis with high-speed railway timetabling: A minimum cycle time calculation model with flexible overtaking constraints and intelligent enumeration. *Transportation Research Part C: Emerging Technologies*, 68, 509-531.
Zhou, X., Zhong, M. 2005. Bicriteria train scheduling for high-speed passenger railroad planning applications. *European Journal of Operational Research*, 167(3), 752-771.
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(3), 320-341.
Zhou, X., Tong, L., Mahmoudi, M., Zhuge, L., Yao, Y., Zhang, Y., Shang, P., Shi, T., 2018. Open-source VRPLite package for vehicle routing with pickup and delivery: a path finding engine for scheduled transportation systems. *Urban Rail Transit*, 4 (2), 68–85.
Zhou, L., Tong, L. C., Chen, J., Tang, J., Zhou, X., 2017. Joint optimization of high-speed train timetables and speed profiles: A unified modeling approach using time-space-speed grid networks. *Transportation Research Part B: Methodological*, 97, 157-181.
Ziliaskopoulos, A.K., Mahmassani, H.S., 1993. Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. *Transportation Research Record*. 94–94.