Expert Systems With Applications 319 (2026) 132033

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Scheduling extra trains to an existing timetable along a railway corridor: An ADMM-based optimisation approach

Jinyi Pan [a], Ronghui Liu [a,*], Zhiyuan Lin [a], Shuguang Zhan [b]

[a] Institute for Transport Studies, University of Leeds, Leeds LS2 9JT, United Kingdom
[b] School of Automotive and Transportation Engineering, Hefei University of Technology, Hefei 230009, China

ARTICLE INFO

Keywords:
Railway timetabling
Extra train insertion
Alternating Direction Method of Multipliers

ABSTRACT

This paper addresses the problem of scheduling extra train paths (SETP) onto an existing timetable along railway corridors. Each extra train has a pre-assigned departure time window: the train can be scheduled to depart at any time within the window, or else a failure to schedule would be registered. Modifications to both existing and extra train paths are permitted, including adjustments to departure and dwell times, the introduction of unplanned stops, and overtaking at stations. The objective function minimises a weighted combination of total travel time of scheduled extra trains and schedule deviations of existing trains. Adjustable weighting factors enable operators to balance punctuality requirements against operational flexibility. We formulate the problem as an Integer Linear Programme (ILP) on a space–time network and develop an Alternating Direction Method of Multipliers (ADMM) decomposition framework. The approach decomposes the problem into train-specific subproblems, each solved using a time-dependent shortest-path algorithm. Coupling constraints are enforced through iterative updates to Lagrange multipliers and penalty parameters. Computational experiments on a real-world corridor demonstrate that the ADMM-based approach achieves feasible solutions within reasonable computation time, with an average optimality gap of 5.43% compared with Lagrangian relaxation lower bounds.

1. Introduction

Train timetabling problems (TTP) involve determining feasible schedules for train movements within a railway network while satisfying a set of operational and safety constraints (Hansen and Pachl, 2014). Solving TTP effectively is fundamental to efficient railway operations and reliable service delivery. Two fundamental TTP tasks exist: (i) train timetable scheduling, which creates a baseline timetable under given infrastructure and operational constraints; and (ii) train timetable rescheduling, which adjusts the timetable to respond to short-term requests for changes and/or to address real-time disruptions or disturbances on the network (Cacchiani et al., 2014; Cacchiani and Toth, 2012; Hansen and Pachl, 2014). Baseline timetables are typically planned months in advance (for instance, six months before implementation in the UK) but may require modifications prior to operation due to engineering work requirements or unexpected demand fluctuations. These modifications range from simply adjusting the departure and/or arrival times of trains to cancelling existing train paths or adding new train paths (Gao et al., 2018; Zheng, 2018).

This paper addresses the problem of scheduling extra train paths (SETP) onto an existing planned timetable. SETP combines elements of both baseline scheduling and rescheduling (Burdett and Kozan, 2009; Tan, 2015): dispatchers must simultaneously generate feasible paths for new trains while managing conflicts with existing services. Unlike real-time rescheduling, which responds dynamically to disruptions (Cacchiani et al., 2014), SETP addresses tactical planning occurring days to weeks before operations, typically driven by anticipated demand increases during holidays, major events, or short-notice freight requests. In SETP, the baseline timetable may be treated as either fixed (non-changeable) or flexible (changeable) (Burdett and Kozan, 2009). This study adopts a flexible approach permitting controlled modifications to existing train schedules. We consider SETP in the context of a general railway corridor with heterogeneous train speeds.

The problem is defined by a baseline timetable for existing trains and a set of extra trains to be inserted. Each extra train is specified by: (i) a departure time window at its origin station, (ii) mandatory stops en-route, (iii) minimum and maximum dwell times at stops, and (iv) running times between consecutive stations. For existing trains, we permit controlled modifications including: (i) adjusting departure times at origin stations within specified bounds, (ii) extending dwell times at

* Corresponding author.
E-mail addresses: tsjp@leeds.ac.uk (J. Pan), R.Liu@its.leeds.ac.uk (R. Liu), Z.Lin@leeds.ac.uk (Z. Lin), shuguangzhan@hfut.edu.cn (S. Zhan).

https://doi.org/10.1016/j.eswa.2026.132033
Received 26 November 2025; Received in revised form 19 February 2026; Accepted 9 March 2026
Available online 19 March 2026

stops, and (iii) introducing unplanned stops or reordering train sequences. All existing trains must be scheduled, whereas extra trains are scheduled only if a feasible path exists within their departure time windows. The objective function minimises a weighted combination of (i) the total travel time of scheduled extra trains and (ii) the schedule deviations of existing trains, with adjustable weighting factors enabling operators to balance punctuality requirements against operational flexibility.

Following the approach of Cacchiani et al. (2010) and Jiang et al. (2017), we formulate SETP as an Integer Linear Programme (ILP) on a space–time network. The large-scale combinatorial nature of SETP renders direct solution computationally intractable for realistic instances. However, decomposition algorithms address this challenge by exploiting the problem's natural structure, where each train's path can be determined independently if conflicts with other trains are temporarily relaxed. We thus propose a decomposition framework based on Alternating Direction Method of Multipliers (ADMM) to solve the problem. Our ADMM framework decomposes SETP into train-specific subproblems, which are solved via time-dependent shortest-path algorithms. Iterative updates to Lagrange multipliers and penalty parameters enforce coupling constraints, such as train headway and station capacity requirements. To the best of our knowledge, this is the first application of ADMM to solve SETP.

The remainder of this paper is organised as follows. Section 2 reviews the existing literature on SETP and the ADMM solution method. Section 3 introduces the problem description and the space–time network representation. Section 4 presents the ILP formulation, including problem objectives and constraints. Section 5 details the ADMM-based solution approach, and the time-dependent shortest-path algorithm used for subproblems. Section 6 provides a case study based on a real-life railway corridor and illustrates the effectiveness of the proposed method. Finally, Section 7 concludes the paper and suggests directions for future research.

2. Literature review

Train scheduling and rescheduling problems have been extensively studied in railway operations research. Interested readers are referred to literature review papers on train scheduling (Cacchiani and Toth, 2012; Parbo et al., 2016; Caimi et al., 2017; Bešinović et al., 2021) and rescheduling (Cacchiani et al., 2014; Corman and Meng, 2015; Gao et al., 2017; Ghaemi et al., 2017; Zhan et al., 2024). By comparison, SETP has received far less attention. We organise the literature review into three subsections: studies on SETP, decomposition methods for train timetabling, and identification of research gaps.

2.1. SETP studies

The SETP addresses the insertion of extra trains into an existing timetable, where the baseline timetable may be either strictly fixed or partially flexible (Burdett and Kozan, 2009; Tan, 2015). The permitted alterations in a SETP problem depend on factors such as the type of trains, the level of timetable utilisation, and the availability of remaining resources (Burdett and Kozan, 2009). In fixed-timetable strategies, common objectives include minimising deviations from ideal requests (Cacchiani et al., 2010), maximising timetable robustness (Flier et al., 2009; Ljunggren et al., 2021), and minimising the total or average travel time of the extra trains (Ingolotti et al., 2004; Lova et al., 2007; Tormos et al., 2008). In practice, when new freight train paths are requested, they are typically scheduled after the passenger timetable has been fixed, reflecting operational priority hierarchies (Cacchiani et al., 2010). Flier et al. (2009) developed a method to predict delay risks when adding freight or passenger trains using regression models based on real-world delay data. Their regression models yielded Pareto-optimal schedules that balance delay risk against travel time, which can aid robust planning. Another notable study by Cacchiani et al. (2010), building on the train scheduling work of Caprara et al. (2002), addressed the SETP for freight trains while maintaining a fixed passenger train timetable. They modelled the problem on a European rail network with alternative routes using a space–time network and developed a heuristic algorithm based on Lagrangian Relaxation (LR) to handle the complex constraints. Freight trains can also be inserted at short notice to meet emergent demand. Such studies still prioritise the passenger schedule, either confining freight trains to the remaining space–time resources (Ljunggren et al., 2021) or allowing only minor adjustments to adjacent train paths (Erlandson et al., 2023).

Flexible strategies are commonly adopted in SETP for passenger services to meet demand, improve connections, or handle emergencies (e.g., Tan, 2015; Jiang et al., 2017; Zheng, 2018). The flexible strategies allow for adjustments to the prescribed timetable by modifying departure/arrival times, running times, stopping times, or even skipping stops (Jiang et al., 2017). SETP can be formulated as a job shop scheduling problem. Burdett and Kozan (2009) developed a three-phase approach combining constructive insertion algorithms and simulated annealing meta-heuristics on a disjunctive graph model, with varying degrees of flexibility for existing train schedules (fixed, semi-fixed, or unfixed). Tan (2013) developed a real-time method for adding new trains, focusing on minimal deviations to the existing timetable. The method decomposed the problem into two sub-problems: optimal insertion for a fixed-order timetable and train reordering. These sub-problems were solved iteratively until no further improvements were possible within a computation time limit. Gao et al. (2018) proposed a three-stage optimisation method to solve the SETP problem: first minimising extra train travel times, then reducing deviations in the existing timetable, and further optimising extra train schedules while keeping the original timetable fixed. Train cancellations were not addressed in this study. Tan (2015) used event-activity networks to formulate the SETP problem as Mixed Integer Programming, where extra train services are inserted into cyclic timetables. While extra trains follow a given frequency, deviations from periodicity are permitted. The model incorporates constraints like headways, station capacity, and periodicity, with objectives to minimise travel time, timetable adjustments, and makespan, while maximising robustness. A branch-and-bound approach with heuristics is employed to solve the problem efficiently.

Several works formulate SETP as ILP models (Cacchiani et al., 2010; Jiang et al., 2017). Jiang et al. (2017) studied the SETP along congested corridors. Their SETP model allows stop-skipping and accounts for acceleration and deceleration times. To solve the problem, constraints such as headways and overtaking were relaxed using LR, with iterative multiplier adjustments for violations. A dynamic programming algorithm is used to determine the shortest train paths on a space–time network. Zheng (2018) addressed scheduling of emergency trains in rail networks, aiming to minimise delays while ensuring minimal disruptions to regular train schedules. It models the problem using a space–time network and introduces a "headroom" function to calculate the capacity for inserting emergency trains. The approach combines a linear algorithm for ideal cases with branch-and-bound procedures for quasi-ideal and feasible solutions, allowing limited schedule adjustments and controlled delays.

2.2. Decomposition methods for solving TTP

Large–scale train scheduling problems are usually computationally challenging due to combinatorial complexity. Decomposition algorithms address this challenge by breaking the problem into smaller, more manageable subproblems that can be solved independently, and therefore have attracted extensive attention. A common decomposition algorithm is the rolling horizon approach, which has been employed to solve train scheduling and rescheduling problems (D'Ariano and Pranzo, 2009; Meng and Zhou, 2019; Zhan et al., 2016), though computational costs can be prohibitive for large-scale instances. Another widely used method is the LR decomposition algorithm, which is a dual ascent method that relaxes complex constraints into objective function, which makes the original problem easier to solve (Caprara et al., 2006, 2002; Hassannayebi et al., 2016; Jiang et al., 2017; Meng and Zhou, 2014). Despite its widespread application, LR faces several challenges. A notable issue is solution symmetry in multi-vehicle problems, which requires careful treatment (Niu et al., 2018). Second, while LR provides strong lower bounds through subgradient methods (Brännlund et al., 1998; Meng and Zhou, 2014), priority-based heuristics often fail to generate feasible upper bound solutions for complex, large-scale instances.

ADMM is another decomposition method that builds on the Douglas-Rachford splitting method (Douglas and Rachford, 1956) and augmented Lagrangian techniques (Hestenes, 1969; Powell, 1969). ADMM was proposed by Glowinski and Marroco (1975) and Gabay and Mercier (1976). Eckstein and Bertsekas (1992) established its theoretical foundation by proving the connection between ADMM and Douglas-Rachford splitting applied to the dual problem. Boyd et al. (2011) provided a comprehensive modern survey of ADMM theory and applications. ADMM combines augmented Lagrangian relaxation with block coordinate descent methods (Bertsekas, 1999). Compared to LR, ADMM augments the objective function with quadratic penalty terms, solving train subproblems sequentially rather than simultaneously via block coordinate descent. This sequential approach offers three key advantages: (i) inherent symmetry breaking (Zhang, 2020); (ii) linearisable quadratic penalties: for binary decision variables, as in train scheduling, the quadratic penalty term can be linearised when other trains are fixed (Zhang et al., 2019; Yao et al., 2019); (iii) simultaneous primal–dual improvement (Boyd et al., 2011; Yao et al., 2019). ADMM has been applied in various transportation optimisation scenarios, including ship trajectory planning (Chen et al., 2018), vehicle routing (Yao et al., 2019), cyclic train timetabling (Zhang et al., 2019), and train rescheduling (Zhan et al., 2022a,b). Recent studies by Xiu et al. (2024) and Gao and Niu (2021) have further extended ADMM by incorporating priority rules for train rescheduling and flexible scheduling (allowing wider departure time windows). The theoretical convergence of applying ADMM to integer programmes is a subtle issue (Zhang et al., 2019). While formal convergence guarantees for integer programmes remain limited, detailed theoretical discussions can be found in Boland et al. (2018) and Yao et al. (2019). Empirical evidence demonstrates that ADMM solutions exhibit good feasibility and quality due to inherent symmetry-breaking properties (Zhang et al., 2019).

2.3. Research gap

Existing SETP studies typically adopt one of three approaches: (i) treating the baseline timetable as strictly fixed with no modifications permitted, limiting extra trains to use residual capacity (Cacchiani et al., 2010; Ljunggren et al., 2021); (ii) sequentially optimising existing and extra trains (Burdett and Kozan, 2009; Tan, 2013; Gao et al., 2018); or (iii) pre-assigning ideal paths to extra trains and applying the same objectives to both existing and extra trains (Cacchiani et al., 2010; Jiang et al., 2017). Among these, Gao et al. (2018) proposed a three-stage optimisation method that permits adjustments to existing train schedules, with comprehensive modelling of acceleration/deceleration times and platform assignment. Similar to Gao et al. (2018), this study permits controlled modifications to existing train schedules. Our contribution focuses on developing an ADMM-based decomposition framework as an alternative solution approach that optimises both train sets simultaneously, rather than sequentially. Additionally, our model permits extra trains to remain unscheduled through an artificial arc mechanism, whereas Gao et al. (2018) require all extra trains to be scheduled.

We apply ADMM to decompose the SETP problem into train-specific subproblems. Compared to Lagrangian Relaxation (Caprara et al., 2006; Cacchiani et al., 2010; Jiang et al., 2017), ADMM incorporates quadratic penalty terms that enhance convergence. Specifically, LR dual solutions frequently violate primal constraints, requiring heuristic repair that may degrade solution quality, whereas ADMM drives solutions toward primal feasibility through quadratic penalties and dynamic parameter adjustment (Xiu et al., 2024). Compared to Rolling Horizon methods that may sacrifice global optimality due to myopic decisions (Zhan et al., 2022b), ADMM maintains the global problem structure. Our approach also differs from prior ADMM applications to cyclic timetabling (Zhang et al., 2019), rescheduling (Zhan et al., 2022a), and flexible scheduling (Gao and Niu, 2021), which optimise a single homogeneous train set. SETP involves two heterogeneous train sets with: (i) differentiated arc costs for calculation of objectives; (ii) differentiated constraint bounds; and (iii) an artificial arc mechanism permitting non-insertion of extra trains. Additionally, our flexible SETP formulation requires only the most basic operational parameters for extra trains (departure time windows, must-stop stations, minimum dwell times, and inter-station running times), enabling practical applicability across diverse operational contexts. Furthermore, we introduce an objective function that explicitly balances two competing goals through adjustable weighting factors: minimising total travel time of scheduled extra trains and minimising deviations to existing train schedules.

3. Problem description and network modelling

3.1. Problem description

This study addresses SETP along a rail corridor. As illustrated in Fig. 1, the set of stations along the corridor is denoted as $V = \{1, ..., s\}$, indexed according to geographical order along the corridor. Stations are categorised into major stations (e.g., stations $1, i, j, s$ where trains can originate or terminate) and intermediate stations (e.g., stations 2, 3, and $s-1$, where trains can only stop or pass through). Each station has capacity C_i, $i \in V$, represented by the number of platforms at the station. The physical link $(i, j) \in E$ denotes the track section between stations i and j. We consider a planning horizon $T = [0, \Theta\delta]$ which is discretised into uniform time intervals of length $\delta = 1$ min, where Θ denotes the total number of time steps. All running, dwell, arrival and departure times are expressed as integer multiples of δ (Gao and Niu, 2021).

The baseline timetable comprises a set K_1 of existing trains and specifies, for each train $k \in K_1$, their: (i) scheduled stopping stations $\overline{V}_k$; (ii) scheduled departure time at origin station and arrival times at intermediate stations; and (iii) a pre-defined dwell time at all scheduled stopping stations. A second set K_2 comprises the extra trains to be inserted. Each $k \in K_2$ is defined by: (i) a departure time window at the origin station; (ii) a list of must-stop intermediate stations; (iii) a minimum dwell time at each stop; (iv) the running time between stations; and (v) the longest permitted travel time. Sectional running times between consecutive stations are assumed fixed according to their speed categories. This assumption implies that acceleration and deceleration times associated with stops are not explicitly modelled. The framework can be extended to capture these effects by constructing multiple running arc types with different durations reflecting stopping patterns at adjacent stations (Jiang et al., 2017). Overtaking is allowed at intermediate stations.

There are two scenarios regarding the existing timetable when scheduling extra trains: (i) keep the baseline timetable fixed, or (ii) allow limited adjustments to it. In this paper, we consider the latter scenario: where the schedule for existing trains is allowed to change within

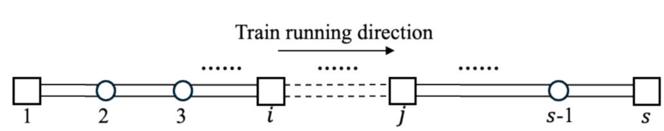

Fig. 1. Schematic of a double-track railway corridor. Stations indexed 1 to s. Rectangles = major stations (trains can originate/terminate); circles = intermediate stations (pass-through or stop only). Arrow indicates train running direction.

specified limits. We illustrate in Fig. 2 the different outcomes of the above two scenarios, when scheduling one slower extra train (Train 4, orange) into a timetable of three existing higher-speed trains (Trains 1–3, blue). Train 4 may depart from S1 anytime during window (D_1, D_2) and it is required to stop at Stations S2 and S3. Let h denote the minimum departure and arrival headway between trains. Fig. 2(a) illustrates the scenario where the paths of all existing (blue) trains remain fixed. To maintain the schedules of the existing trains and have a conflict-free schedule for Train 4, long dwell times must be set for Train 4 at stations S2 and S3. Fig. 2(b) illustrates the scenario where the original schedule (dotted lines) of the existing trains is allowed to alter. The altered timetable (solid lines) shows a slightly later departure of existing Train 1, and slightly longer dwell time of Train 2 at station S2 and Train 3 at station S3, while the inserted Train 4 has a much shorter dwell time at stations S2 and S3 and shorter overall journey time.

Following Jiang et al. (2017) and Gao et al. (2018), we allow existing train paths to deviate within specified limits: (i) the departure time of train $k \in K_1$ at origin station can move earlier or later up to sh_k; (ii) they can make additional stops en-route and (iii) while keeping a minimum dwell time equal to that in their original schedule, train k may extend its dwell time (at originally scheduled and additional stops) by up to a maximum dwell time. These tight modification constraints for existing trains reflect their higher operational priority compared to extra trains. Scheduling for the extra train $k \in K_2$ will determine: (i) its departure time from origin station within a pre-defined and typically wider departure time window; (ii) stopping patterns which will include the must-stop stations and additional stops; (iii) dwell time subject to a maximum that is higher than that for the existing trains. The objective function minimises the total travel times of all scheduled extra trains. For any extra train that cannot be scheduled within its departure time window or whose travel time exceeds the predefined maximum threshold, a penalty cost is incurred in the objective function, whereas all existing trains must be scheduled (enforced by a large penalty).

Hence, the SETP problem addressed in this paper is defined as follows:

SETP definition. Given a baseline timetable for K_1 and extra train requests K_2, SETP determines feasible paths that accommodate as many extra trains as possible while permitting controlled modifications to existing schedules. A feasible path specifies departure times, stopping stations, and dwell times, satisfying station capacity and headway constraints. The objective minimises a weighted sum of existing train deviations and extra train travel times.

Table 1 lists the relevant indices, sets, parameters and decision variables used in this paper to formulate the SETP model.

3.2. Space-time network

Space-time network (a graph representation where nodes encode station-time pairs and arcs represent train movements) is widely used in TTP models because of its ability to capture the spatiotemporal characteristics of a timetable and transforms complex constraints (e.g., reachability, incompatibility, and flow conservation) into simple restrictions on arcs or nodes (Gao and Niu, 2021; Xiu et al., 2024). In a space–time network $G = (N, A)$, N represents space–time nodes and A contains space–time arcs. Specifically, a physical station node $i \in V$ is expanded into a space node $(i, t) \in N$, and a physical section $(i, j) \in E$ is expanded into a space–time arc $(i, t, j, t') \in A$; the latter indicates that a train leaving node i at time t would arrive at node j at time t'. For flow conservation, each train $k \in K_1 \cup K_2$ is assigned a virtual origin node $\bar{o}_k$ and a virtual destination node $\bar{d}_k$. Consequently, the node set N includes station nodes as well as $\bar{o}_k$ and $\bar{d}_k$ for each train.

In a space–time network, trains traverse various arc types, representing different operational states (Zhan et al., 2022a). As illustrated in Fig. 3, these include:

- a run arc $(i, t, j, t') \in A^{run}$ which represents a train departing station i at time t and arriving at station j at time t'. The duration $(t' - t)$ is the travel time.
- a dwell arc $(i, t, i, t') \in A^{dwell}$ which represents a train's scheduled stop at station i from t to t'. The duration $(t' - t)$ is the minimum dwell time.
- an extra dwell arc $(i, t, i, t + \Delta t_{stop}) \in A^{stop}$ which represents extended dwell times or dwell time at an unplanned stop at station i, for Δt_{stop} minutes.
- an access arc $(\bar{o}_k, t, o_k, t') \in A^{access}$ which connects the train's virtual origin $\bar{o}_k$ to its first station o_k, while an egress arc $(d_k, t, \bar{d}_k, t') \in A^{egress}$ which connects the last station d_k to the virtual destination $\bar{d}_k$.
- an artificial arc $(\boldsymbol{\bar{o}_k}, \boldsymbol{e_k}, \boldsymbol{\bar{d}_k}, \boldsymbol{l_k}) \in \boldsymbol{A^{artificial}}$ which is used when a train cannot be scheduled. $\boldsymbol{e_k}$ and $\boldsymbol{l_k}$ are train $\boldsymbol{k}$'s earliest departure time from $\boldsymbol{\bar{o}_k}$ and latest arrival time at $\boldsymbol{\bar{d}_k}$, respectively.

Train travel times and minimum dwell times are inherently enforced by the network construction. Note that although the running time between two consecutive stations is considered fixed in this study, the

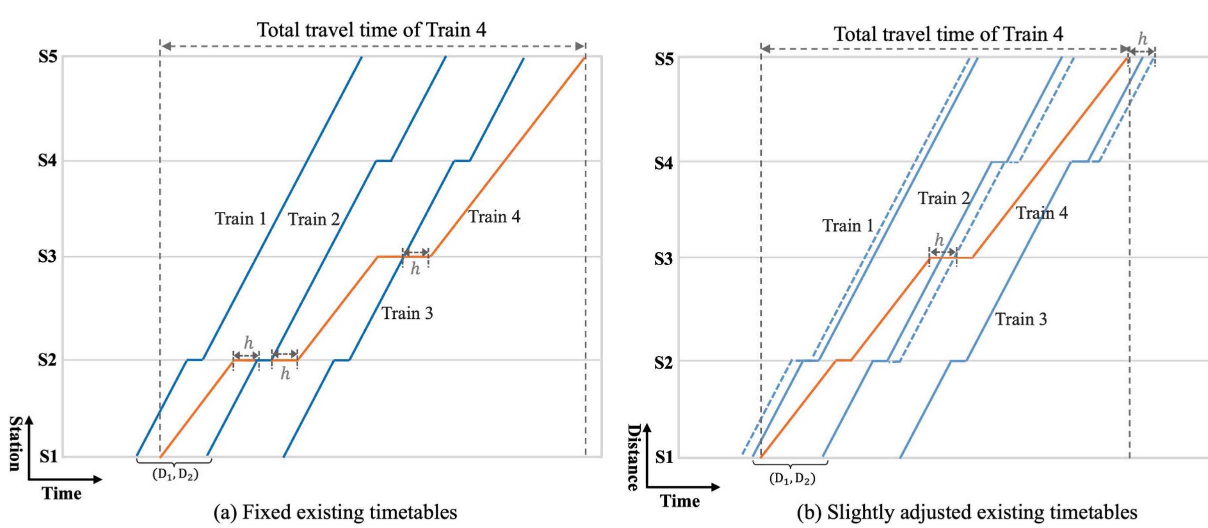

Fig. 2. Comparison of fixed and flexible timetabling strategies for extra train insertion. (a) Fixed strategy: existing trains (blue) maintain original schedules; Train 4 (orange) experiences longer dwell times. (b) Flexible strategy: minor adjustments to existing trains (dotted lines) enable earlier departure and shorter travel time for Train 4. $\boldsymbol{h}$ represents minimum arrival/departure headways.

Table 1
Definition of indices, sets, parameters and decision variables.

Indices	Definition
k, k'	Index of trains, $k, k' \in K$
i, j	Index of stations, $i, j \in V$
(i, j)	Track section between stations i and j, $(i, j) \in E$
t, t', τ, τ'	Time indices in the space–time network, $t, t', \tau, \tau' \in T$
(i, t), (j, t')	Index of space–time nodes, (i, t), $(j, t') \in N$
(i, t, j, t')	Index of space–time arcs, $(i, t, j, t') \in A$
a	Compact notation for arc (i, t, j, t')
Sets	
K	Set of trains
K_1	Set of existing trains, $K_1 \subset K$
K_2	Set of requested extra trains, $K_2 \subset K$
V	Set of stations
V_k	Set of stations traversed by train k, $V_k \subseteq V$
$\overline{V}_k$	Subset of mandatory stops for train k, $\overline{V}_k \subseteq V_k$
E	Set of physical sections
T	Set of discrete times of the planning horizon
N	Set of space–time nodes
N_k	Set of space–time nodes for train k
A	Set of space–time arcs
A_k	Set of space–time arcs for train $k \in K$, $A_k \subset A$
$U(i, t, j, t')$	Set of arcs that are incompatible with space–time arc (i, t, j, t')
Parameters	
o_k	Origin station of train k
d_k	Destination station of train k
$\overline{o}_k$	Dummy origin node of train k in the space–time network
$\overline{d}_k$	Dummy destination node of train k in the space–time network
Δt_{stop}	Duration of each extra dwell arc (in minutes)
$(DT_k{}^e, DT_k{}^l)$	The earliest and latest departure time of train k at its origin station, $k \in K_2$
sh_k	Maximum allowable departure time shift (in minutes) for existing train $k \in K_1$ at its origin station
$c^k_{itjt'}$	Deviation cost of the space–time arc (i, t, j, t') for train $k \in K_1$
$\tilde{c}^k_{itjt'}$	Travel cost of the space–time arc (i, t, j, t') for train $k \in K_2$
C_i	Capacity of station $i \in V$
$tw^{min}_{k,i}$	Minimum dwell time of train k at station $i \in V_k \backslash \{o_k, d_k\}$
$tw^{max}_{k,i}$	Maximum dwell time of train k at station $i \in V_k \backslash \{o_k, d_k\}$
h^{arr}	Safe arrival headway between consecutive trains to the same station
h^{dep}	Safe departure headway between consecutive trains from the same station
$ta_{k,i}$	Scheduled arrival time of train k to station $i \in V_k$
$td_{k,i}$	Scheduled departure time of train k from station $i \in V_k$
ω_1	Weight of deviation cost of existing trains
ω_2	Weight of travel cost of extra trains
Decision variables	
$x^k_{i,t,j,t'}$	If train k uses arc (i, t, j, t'), then $x^k_{i,t,j,t'} = 1$; otherwise, $x^k_{i,t,j,t'} = 0$

above method allows adding a running arc to represent variability in travel time. For a train k planned to stop at station i, if the arc preceding the node (i, t) is a running arc, the subsequent arc (i, t, j, t') must be a dwell arc. Conversely, if the train is not scheduled to stop at i, the subsequent arc must either be a running arc (passing i) or an extra dwell arc (making an unplanned stop at i). Extra dwell arcs are available at all intermediate stations for every train, and the selection between passing and stopping is determined by the shortest path algorithm in the ADMM subproblem solver (Algorithm 2). If no feasible path is found for an extra train k, an artificial arc is assigned to the train.

4. Mathematical formulation

4.1. Decision variables

For each $k \in K_1 \cup K_2$ and its available space–time arc $(i, t, j, t') \in A_k$, let the binary variable $x^k_{i,t,j,t'} = 1$ if and only if arc $(i, t, j, t') \in A_k$ is chosen, and $x^k_{i,t,j,t'} = 0$, otherwise. This yields the domain constraint:

$$x^k_{i,t,j,t'} \in \{0, 1\} \forall k \in K_1 \cup K_2, \forall (i, t, j, t') \in A_k \tag{1}$$

These variables identify the train paths in the final schedule.

4.2. Objective function

Using the binary arc-based variables defined above, the SETP problem can be expressed as a multi-commodity network flow model, where each commodity represents a complete train path from its origin to its destination through the space–time network. The representation of arc costs varies depending on the arc type and train type. For existing trains $k \in K_1$, the objective is to minimise the total arrival time deviations from the original timetable, i.e., the sum of arrival time deviation at each station and over all trains. For existing trains, only running arcs contribute to deviation costs, defined as $c^k_{i,t,j,t'} = |t' - ta_{k,j}|$, where $ta_{k,j}$ is the scheduled arrival time of train k at station j. Dwell arcs, extra dwell arcs, access arcs, and egress arcs have zero deviation cost ($c^k_{i,t,j,t'} = 0$). For extra trains $k \in K_2$, the objective is to minimise total travel time, with the costs for running, dwell, and extra dwell arcs determined by their associated transition times, i.e., for a running arc $(i, t, j, t') \in A^{run}$ or a dwell arc $(i, t, i, t') \in A^{dwell}$, the arc cost $\tilde{c}^k_{i,t,j,t'}$ is set as $t' - t$, while the cost of an extra dwell arc $(i, t, i, t + \Delta t_{stop}) \in A^{stop}$ is Δt_{stop}. For existing trains, a very high cost $R^{existing}_k$ is assigned to artificial arcs to prevent their cancellation. For an extra train, a cost R^{extra}_k is applied to artificial arcs, which equals the longest travel time; if no feasible path exists within this limit, the train selects the artificial arc, contributing R^{extra}_k to the objective function as a penalty for non-scheduling.

The objective function then minimises a weighted sum of these two components:

$$\min Z = \omega_1 \sum_{k \in K_1} \sum_{(i,t,j,t') \in A_k} c^k_{i,t,j,t'} \times x^k_{i,t,j,t'}$$

$$+ \omega_2 \sum_{k \in K_2} \sum_{(i,t,j,t') \in A_k} \tilde{c}^k_{i,t,j,t'} \times x^k_{i,t,j,t'} \tag{2}$$

The first term represents the total arrival deviation cost for existing trains, while the second term captures the total travel time cost for extra trains. The weighting factors ω_1 and ω_2 can be chosen by the operator to balance existing timetable's punctuality against extra trains' travel times. Note that alternative SETP formulations maximise total profit, where each scheduled train contributes positive profit (Cacchiani et al., 2010; Jiang et al., 2017). Our cost-minimisation formulation achieves similar scheduling incentives through the artificial arc penalty R^{extra}_k, while optimising travel time and deviation for scheduled trains.

4.3. Constraints

1) Flow conservation constraints

The flow conservation constraint guarantees that each train $k \in K_1 \cup K_2$ has exactly one path from its virtual origin $\overline{o}_k$ to destination $\overline{d}_k$:

$$\sum_{(j,t'):(i,t,j,t') \in A_k} x^k_{i,t,j,t'} - \sum_{(j,t'):(j,t',i,t) \in A_k} x^k_{j,t',i,t} = \begin{cases} 1 i = \overline{o}_k, t = e_k \\ -1 i = \overline{d}_k, t = l_k \forall k \in K_1 \cup K_2 \\ 0 otherwise \end{cases} \tag{3}$$

Remark: At scheduled stops, the arc-sequence rule (a dwell arc must immediately follow a running arc) is not modelled as an explicit ILP constraint but is enforced by the predecessor-tracking logic in the dynamic programming subproblem (Zhan et al., 2022a).

2) Station capacity constraint

Constraint (4) ensures that at any time t, the total number of trains at station i does not exceed the station capacity C_i. The constraint consists of two summation terms corresponding to different train activities (Zhan et al., 2022a).

The first term counts trains occupying platforms while dwelling at station i. This includes dwell arcs $(i, \tau, i, t') \in A^{dwell}_k$, which represent

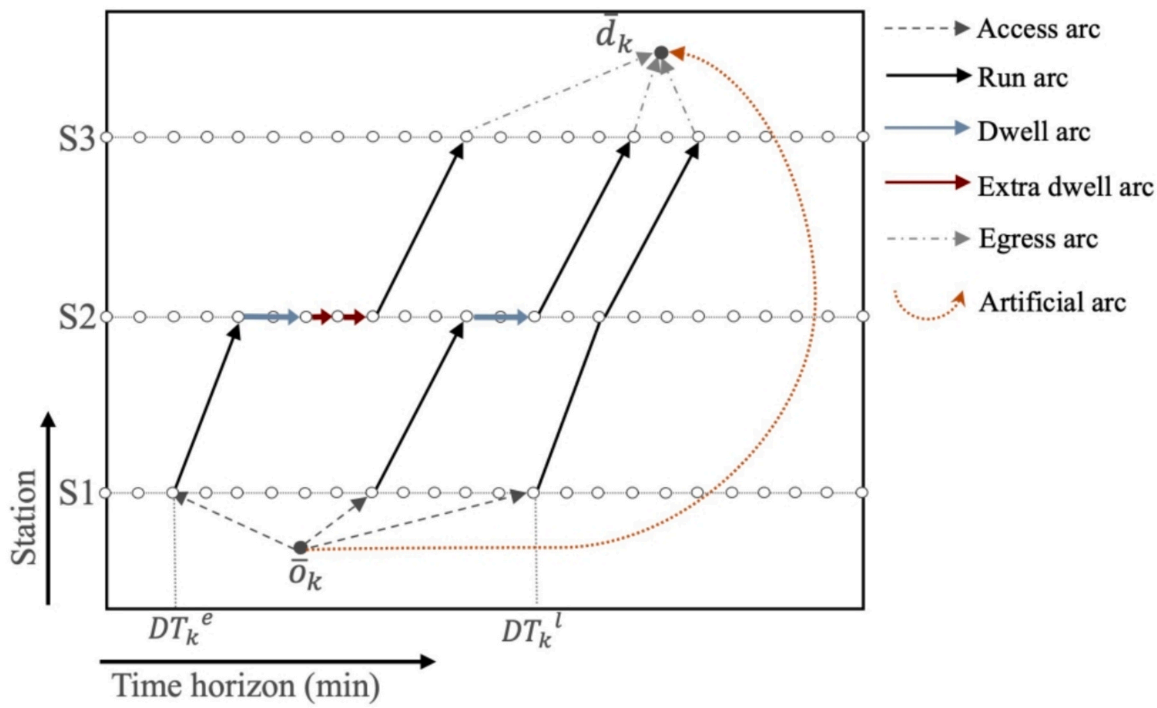

Fig. 3. Arc types in space–time network. A train's path from virtual origin $\bar{o}_k$ to virtual destination $\bar{d}_k$ through three stations (S1, S2, S3). Vertical axis: stations; horizontal axis: time ($DT_k{}^e$ = earliest departure, $DT_k{}^l$ = latest departure). Arc types shown in legend: (1) Access arc (dashed): connects $\bar{o}_k$ to physical origin o_k; (2) Run arc (solid diagonal): travel between stations; (3) Dwell arc (solid horizontal, blue): mandatory stop; (4) Extra dwell arc (dashed horizontal, red): extended or unplanned stop; (5) Egress arc (dashed): connects destination d_k to $\bar{d}_k$; (6) Artificial arc (curved): used when train cannot be feasibly scheduled.

trains whose scheduled dwelling period $\left[\tau, \tau + tw_{k,i}^{min}\right]$ covers time t, i.e., those with $t - tw_{k,i}^{min} < \tau \le t$, as well as extra dwell arcs $(i,t,i,t') \in A_k^{stop}$, which capture trains beginning extended or unplanned stops at time t (where $\tau = t$). The strict inequality $t - tw_{k,i}^{min} < \tau$ ensures mutual exclusivity: at the transition from dwelling to departing ($t = \tau + tw_{k,i}^{min}$), the dwell arc no longer contributes, while the departing run arc is counted. Thus, each train contributes to exactly one term at any time t. For extra dwell arcs, τ equals t in the summation. The second term counts trains departing from station i at time t through run arcs $(i,t,j,t') \in A_k^{run}$, where $j = i+1$ is the next station.

This constraint is expressed as:

$$\sum_{k\in K_1\cup K_2}\sum_{\tau\in T}\sum_{(i,\tau,i,t')\in A_k} x_{i,\tau,i,t'}^k + \sum_{k\in K_1\cup K_2}\sum_{(i,t,j,t')\in A_k} x_{i,t,j,t'}^k$$

$$\le C_i \forall i \in V : j = i+1, \forall t \in T \tag{4}$$

3) Maximum dwell time

When a train stops at an intermediate station i, it uses a dwell arc with a duration of $tw_{k,i}^{min}$, ensuring sufficient time for operations such as boarding and alighting. To prevent excessive waiting, its total dwell time at station i cannot exceed $tw_{k,i}^{max}$. The additional dwell time available beyond the minimum requirement is therefore $tw_{k,i}^{max} - tw_{k,i}^{min}$. Summing over all such arcs yields the total extra dwell time at station i for train k. We then enforce:

$$\sum_{(i,t,i,t')\in A^{stop}} x_{i,t,i,t'}^k \times \Delta t_{stop} \le tw_{k,i}^{max} - tw_{k,i}^{min} \forall i \in V_k \backslash \{o_k, d_k\}, \forall k \in K_1 \cup K_2 \tag{5}$$

4) Train headway and overtaking constraints

To ensure operational safety between consecutive trains, train headway constraints must be maintained. Additionally, allowing trains with different speeds to run within a segment enables overtaking. This study addresses headway and overtaking constraints using incompatible sets of arcs, building on methods from previous research (Xiu et al., 2024; Zhan et al., 2022a). For notational simplicity, when referring to an arc $a = (i,t,j,t')$, we write x_a^k to denote the decision variable $x_{i,t,j,t'}^k$, and c_a^k (or $\tilde{c}_a^k$) to denote the arc cost $c_{i,t,j,t'}^k$ (or $\tilde{c}_{i,t,j,t'}^k$). For a run arc $a = (i,t,j,t') \in A^{run}$, let $U(a)$ denote the set of all conflicting run arcs $a' = (i,\tau,j,\tau') \in A^{run}$ on the same section (i,j) in the space–time network. Fig. 4 illustrates two cases of headway incompatibility, in which arc a represents a high-speed arc and a low-speed arc, respectively. In Fig. 4 (a), a is the left boundary of $U(a)$. Arc a' is incompatible with a if their departure time difference is less than the minimum departure headway h^{dep} or $(t' - t) - (\tau' - \tau) + h^{arr}$, where $(t' - t)$ and $(\tau' - \tau)$ are the running times of the two run arcs. Let $h_{a,a'} = \max\{h^{dep}, [(t' - t) - (\tau' - \tau) + h^{arr}]$, and the corresponding incompatibility set becomes $\{a' \in A^{run} | t \le \tau < t + h_{a,a'}\}$. Fig. 4(b) illustrates similar cases where a is the right boundary of set $U(a)$. In these cases, arc a′ is incompatible with arc a if their departure time difference is less than h^{dep} or $(\tau' - \tau) - (t' - t) + h^{arr}$. In addition, we define a parameter $h_{a',a}$ = max $\{h^{dep}, (\tau' - \tau) - (t' - t) + h^{arr}\}$, and the resulting incompatibility set is $\{a' \in A^{run} | t - h_{a',a} \le \tau < t\}$. Based on these two scenarios, the headway incompatibility set $U(a)$ for arc a is given by:

$$U(a) = \{a' = (i,\tau,j,\tau') \in A^{run} : t - h_{a',a} \le \tau < t + h_{a,a'}\} a = (i,t,j,t') \in A^{run} \tag{6}$$

Consider a high-speed arc a with running time 10 min departing at $t = 5$, and a low-speed arc a' with running time 12 min departing at τ, with $h^{dep} = h^{arr} = 3min$. Applying Eq. (6): $h_{a,a'} = max\{3, (10-12)+3\} = 3min$ and $h_{a',a} = max\{3, (12-10)+3\} = 5min$, yielding $U(a) = \{a' : 0 \le \tau < 8\}$. If the low-speed train departs at $\tau = 1$, it arrives at $\tau' = 13$, while the high-speed train arrives at $t' = 15$, giving arrival separation $15 - 13 = 2 < h^{arr}$—a conflict correctly captured by the incompatibility set.

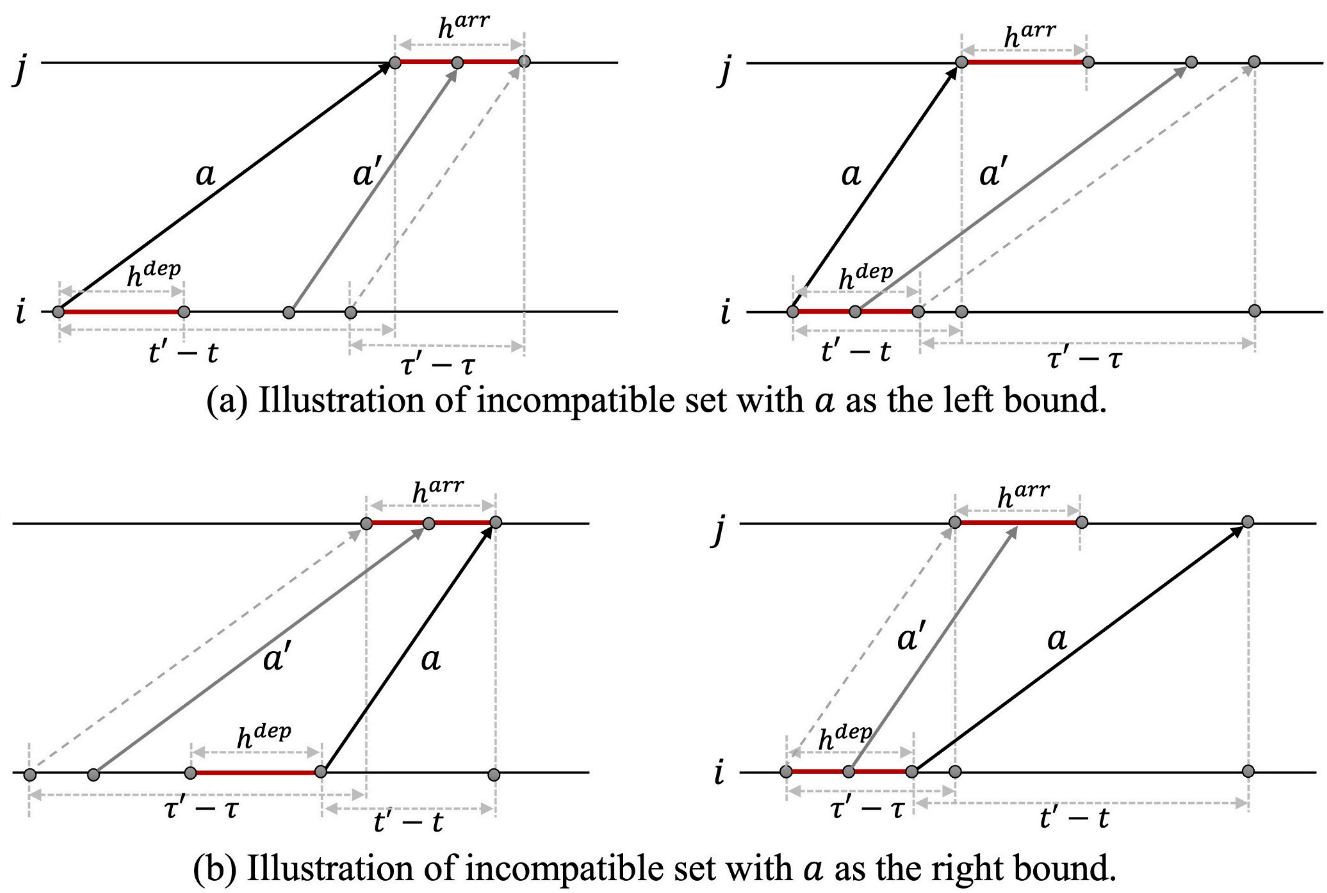

Fig. 4. Incompatibility sets for headway constraints. Four cases illustrating headway incompatibility. Arcs $\boldsymbol{a}$ (solid) and $\boldsymbol{a}'$ (dashed) on section $(\boldsymbol{i},\boldsymbol{j})$ are incompatible when they violate minimum headways. Red shaded regions = conflicting departure times for $\boldsymbol{a}'$. (a) Arc $\boldsymbol{a}$ as left boundary (b) Arc $\boldsymbol{a}$ as right boundary. $\boldsymbol{h}^{arr}$, $\boldsymbol{h}^{dep}$ = minimum arrival/departure headways.

With the incompatibility set defined above, the corresponding constraint is formulated as follows:

$$\sum\nolimits_{k'\in K_1\cup K_2}\sum\nolimits_{a\in A_{k'}}\boldsymbol{x}_{\boldsymbol{a}}^{\boldsymbol{k}'}+\sum\nolimits_{a'\in U(a)}\boldsymbol{x}_{\boldsymbol{a}'}^{\boldsymbol{k}}\leq 1\forall a$$

$$=(\boldsymbol{i},\boldsymbol{t},\boldsymbol{j},\boldsymbol{t}')\in \boldsymbol{A}^{run}: i\neq j, \forall k\in \boldsymbol{K}_1\cup \boldsymbol{K}_2 \tag{7}$$

The above incompatibility constraints are compact, relying only on a single train index and arc index (Gao and Niu, 2021). They handle speed differences through timing adjustments rather than explicit overtaking variables and can apply to general mixed-speed operations. For each train $\boldsymbol{k}\in \boldsymbol{K}_1\cup \boldsymbol{K}_2$, the construction of the available arc set $\boldsymbol{A}_{\boldsymbol{k}}$ implicitly ensures adherence to the prescribed travel times between successive stations and the minimum required dwelling times at all scheduled stops. Moreover, the departure time window constraints are incorporated for each train, as will be detailed in Section 5.

5. Decomposition framework based on ADMM

The SETP problem formulated in Section 4 results in a large-scale ILP. Whilst the number of binary variables $\boldsymbol{x}_{i,t,j,t'}^{\boldsymbol{k}}$ is bounded by $|\mathrm{K}| \times |\mathrm{E}| \times |\mathrm{T}|$, the number of constraints is also substantial. For example, the numbers of constraints required for flow conservation constraint (3), safety headway (7) and station capacity (4) are respectively $|\boldsymbol{K}|\times(|\boldsymbol{V}|-2)\times|\boldsymbol{T}|+2|\boldsymbol{K}|$, $|\boldsymbol{K}|\times|\boldsymbol{E}|\times|\boldsymbol{T}|$, and $|\boldsymbol{V}|\times|\boldsymbol{T}|$. For realistic instances such as K = 94 trains, E = 10 track sections and T = 720 timesteps, these yield approximately 677,000 variables and over 1.3 million constraints, making the problem intractable for commercial ILP solvers. From a mathematical programming perspective, the formulated ILP naturally yields a block-diagonal structure. Each train's shortest path subproblem constitutes a distinct block, while additional coupling constraints link these blocks (Zhang et al., 2019). Therefore, although commercial ILP solvers can handle small instances, large-scale instances remain computationally challenging as conflicts over shared track segments and station platforms must be resolved. To address these challenges, various dual decomposition strategies have been proposed to exploit problem structure and reduce computational effort (Bertsekas, 1997; Boyd et al., 2011). Lagrangian relaxation, for instance, integrates complex constraints into the objective function as penalty terms, decomposing the large-scale problem into smaller, more tractable subproblems. Augmented LR further introduces quadratic penalties to improve numerical stability and convexity, though it complicates parallel updates of variables. ADMM combines augmented LR with block-coordinate descent (Bertsekas, 1997). By updating subsets of variables in an alternating manner, ADMM retains desirable decomposition properties while enhancing robustness and convergence characteristics. While we adopt a similar ADMM framework from prior work (Zhang et al., 2019; Zhan et al., 2022a), the heterogeneous nature of SETP—with two train sets having different objectives and constraints—requires specific adaptations in arc cost calculations (Eq. (20) and subproblem formulations (Algorithm 2).

An illustrative overview of the ADMM solution process is presented in Fig. 5. The ADMM framework consists of four main steps, detailed as follows: we first relax the original SETP problem into a LR model and then add a quadratic term based on the structure of the incompatibility sets. Once this term is linearised, the augmented model decomposes SETP into individual train subproblems, each solvable in turn by a time-dependent shortest-path algorithm.

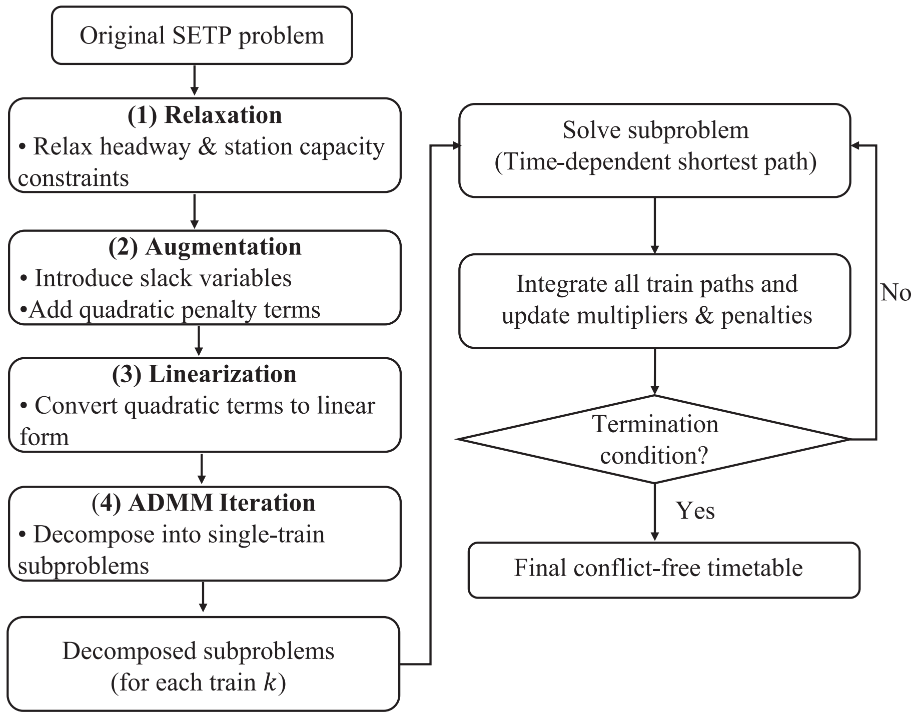

Fig. 5. ADMM-based decomposition framework for SETP. Left: Four-stage transformation: (1) Relaxation of coupling constraints; (2) Augmentation with slack variables; (3) Linearization of quadratic terms; (4) Decomposition into train subproblems. Right: Iterative process including solve subproblems via shortest path, integrate solutions, update multipliers/penalties, check termination condition. Repeats until maximum iterations reached.

(1) Relaxation

In the first step, we relax constraints that link multiple trains. As the headway and overtaking constraints (7) and the station capacity constraints (4) collectively involve many trains and complicate the original problem, we address these problems following Zhan et al. (2022a). More specifically, for the station capacity constraint (4), which ensures that the total number of trains at station $\boldsymbol{i}$ at any time does not exceed its capacity $\boldsymbol{C_i}$, we introduce a Lagrangian multiplier γ_i^t for each station $\boldsymbol{i}$ at each time $\boldsymbol{t}$. This transformation shifts the capacity constraint into the objective function by penalising any violation proportional to γ_i^t. Similarly, for the headway and overtaking constraints (7), which require that the time difference between conflicting run arcs is no less than $\boldsymbol{h_{a,a'}}$, we introduce a Lagrangian multiplier λ_a^k for each run arc $\boldsymbol{a}$. This transformation incorporates a penalty for any simultaneous selection of conflicting arcs that would violate the headway requirement.

After applying these multipliers, the LR model can be expressed as follows:

$$\boldsymbol{minZ_L} = \omega_1 \times \left(\sum\nolimits_{k\in K_1}\sum\nolimits_{a\in A_k} c_a^k \times x_a^k\right) + \omega_2 \times \left(\sum\nolimits_{k\in K_2}\sum\nolimits_{a\in A_k} \tilde{c}_a^k \times x_a^k\right) + \sum\nolimits_{a\in A^{run}}\sum\nolimits_{k\in K_1\cup K_2}\lambda_a^k\left(\sum\nolimits_{k'\in K_1\cup K_2}\sum\nolimits_{a\in A_{k'}} x_a^{k'} + \sum\nolimits_{a'\in U(a)} x_{a'}^k - 1\right) + \sum\nolimits_{i\in V}\sum\nolimits_{t\in T}\gamma_i^t\left(\sum\nolimits_{k\in K_1\cup K_2}\sum\nolimits_{\tau\in T}\sum\nolimits_{a''\in A^{dwell}\cup A^{stop}} x_{a''}^k + \sum\nolimits_{k\in K_1\cup K_2}\sum\nolimits_{a\in A^{run}} x_a^k - C_i\right) \tag{8}$$

Subject to constraints (1), (3) and (5).

The mutual covering structure of incompatible arc sets $\boldsymbol{U(a)}$ allows the penalty terms to be rearranged. Specifically, for any train $\boldsymbol{k}$, the following equivalence holds (proof in Appendix C):

$$\sum\nolimits_a \lambda_a^k \sum\nolimits_{a'\in U(a)} x_{a'}^k = \sum\nolimits_{a'\in U(a)} \lambda_{a'}^k \times \sum\nolimits_a x_a^k \tag{9}$$

To illustrate, consider a single train $\boldsymbol{k}$ traversing the network. Suppose that within segment $(\boldsymbol{i},\boldsymbol{j})$, train $\boldsymbol{k}$ uses exactly one arc $\boldsymbol{a_0} \in \boldsymbol{U(a)}$, i.e., $x_{a_0}^k = 1$ and $x_a^k = 0$ for $\boldsymbol{a} \neq \boldsymbol{a_0}$. Then we have:

$$\sum\nolimits_a \lambda_a^k \sum\nolimits_{a'\in U(a)} x_{a'}^k = \sum\nolimits_a \lambda_a^k \left(\sum\nolimits_{a_0\in U(a)} x_{a_0}^k + 0\right)$$

$$= \sum\nolimits_{a\in U(a_0)} \lambda_a^k x_{a_0}^k = \sum\nolimits_{a'\in U(a)} \lambda_{a'}^k \times \sum\nolimits_a x_a^k \tag{10}$$

Using Eq.(9) and equality $\sum\nolimits_{K\in K_1\cup K_2}\sum\nolimits_{a\in A^{run}}\lambda_a^k \times \sum\nolimits_{k'\in K_1\cup K_2} x_a^{k'} = \sum\nolimits_{K\in K_1\cup K_2}\sum\nolimits_{a\in A^{run}} x_a^k \times \sum\nolimits_{k'\in K_1\cup K_2}\lambda_a^{k'}$ we therefore decompose LR into independent single-train sub-problems. Letting Z_L^k denote the Lagrangian objective function for train $\boldsymbol{k}$, we have:

$$Z_L^k = \omega_1 \times \sum_{a\in A_k} c_a^k \times x_a^k + \omega_2 \times \sum_{a\in A_k} \tilde{c}_a^k \times x_a^k + \sum_{a\in A^{run}} x_a^k \times \sum_{k'\in K_1\cup K_2} \lambda_a^{k'} + \sum_{a\in A^{run}} x_a^k \times \sum_{a'\in U(a)} \lambda_{a'}^k + \sum_{i\in V}\sum_{t\in T}\gamma_i^t \times \left(\sum_{\tau\in T}\sum_{a''\in A^{dwell}\cup A^{stop}} x_{a''}^k + \sum_{a\in A^{run}} x_a^k\right) = \dot{c}_a^k \times x_a^k \tag{11}$$

$\dot{c}_a^k$ represents the modified arc cost, incorporating the effects of Lagrangian multipliers. See Appendix A for more details of $\dot{c}_a^k$. At this point, the LR model is decomposed into a set of independent single-train subproblems, each with a linear objective function.

(2) Augmentation

To improve convergence properties of the LR model, we augment the objective function with quadratic penalty terms. Following the approach described in previous studies (Xiu et al., 2024; Gao and Niu, 2021), we introduce two nonnegative slack variables, $\pi(\boldsymbol{a})$ and $\mu(\boldsymbol{i})$, both with a range of $[0,1]$, to rewrite the consensus constraints (4) and (7) in a more tractable form:

$$\sum_{k'\in K_1\cup K_2}\sum_{a\in A_{k'}} x_a^{k'} + \sum_{a'\in U(a)} x_{a'}^k - 1 + \pi(\boldsymbol{a}) = 0 \forall a\in \boldsymbol{A}^{run}, k\in \boldsymbol{K}_1\cup \boldsymbol{K}_2 \tag{12}$$

$$\sum_{k\in K_1\cup K_2}\sum_{\tau\in T}\sum_{a''\in A^{dwell}\cup A^{stop}} x_{a''}^k + \sum_{k\in K_1\cup K_2}\sum_{a\in A^{run}} x_a^k - C_i + \mu(\boldsymbol{i}) = 0 \forall i\in V, \forall t\in T \tag{13}$$

To further penalise violations of these constraints, we add quadratic penalty terms with parameters ρ_1 and ρ_2, extending the LR model to an Augmented LR model:

$$Z_{AL}^k = Z_L^k + \frac{\rho_1}{2}\sum_{a\in A^{run}}\sum_{k\in K}\left\|\sum_{k'\in K_1\cup K_2}\sum_{a\in A_{k'}} x_a^{k'} + \sum_{a'\in U(a)} x_{a'}^k - 1 + \pi(\boldsymbol{a})\right\|_2^2 + \frac{\rho_2}{2}\sum_{i\in V}\sum_{t\in T}\left\|\sum_{k\in K_1\cup K_2}\sum_{\tau\in T}\sum_{a''\in A^{dwell}\cup A^{stop}} x_{a''}^k + \sum_{k\in K_1\cup K_2}\sum_{a\in A^{run}} x_a^k - C_i + \mu(\boldsymbol{i})\right\|_2^2 \tag{14}$$

where $\|\bullet\|^2$ is the squared Euclidean norm; the two squared terms respectively penalise (i) the headway and overtaking constraints (7) and (ii) the station capacity constraint (4).

(3) Linearisation

Incorporating these quadratic terms directly makes the model non-decomposable. By utilising binary decision variables and the known schedules of other trains, we can transform these quadratic expressions into linear ones. Specifically, we introduce auxiliary variables that represent fixed occupation of arcs and stations determined by all other trains. Let φ_a^k denote the total occupation of arc $\boldsymbol{a}$ and its incompatible arcs by all trains other than the current train $\boldsymbol{k}$:

$$\varphi_a^k = \sum_{k'\in K_1\cup K_2/\{k\}}\sum_{a\in A_{k'}} x_a^{k'} + \sum_{k'\in K_1\cup K_2\backslash\{k\}}\sum_{a'\in U(a)} x_{a'}^{k'} \tag{15}$$

Similarly, let $\zeta_{i,t}^k$ denote the total station occupation of station $\boldsymbol{i}$ at time $\boldsymbol{t}$ by all trains other than train $\boldsymbol{k}$:

$$\zeta_{i,t}^k = \sum_{k'\in K_1\cup K_2/\{k\}}\sum_{\tau\in T}\sum_{a''\in A^{dwell}\cup A^{stop}} x_{a''}^{k'} + \sum_{k'\in K_1\cup K_2/\{k\}}\sum_{a\in A^{run}} x_{a''}^{k'} \tag{16}$$

Because the paths of other trains are fixed when scheduling train $\boldsymbol{k}$, φ_a^k and $\zeta_{i,t}^k$ are known constants when at that stage. By substituting these constants into the objective function, we can replace the quadratic terms with linear expressions. To illustrate the linearisation process, we take the first quadratic term involving $\pi(\boldsymbol{a})$ as example:

$$\begin{aligned}
&\frac{\rho_1}{2}\sum_{a\in A^{run}}\sum_{k\in K_1\cup K_2}\left\|\sum_{k'\in K_1\cup K_2}\sum_{a\in A_{k'}} x_a^{k'} + \sum_{a'\in U(a)} x_{a'}^k - 1 + \pi(\boldsymbol{a})\right\|_2^2\\
&= \frac{\rho_1}{2}\sum_{a\in A^{run}}\sum_{k\in K_1\cup K_2}\left\|x_a^k + \sum_{k'\in K_1\cup K_2/\{k\}}\sum_{a\in A_{k'}} x_a^{k'} + \sum_{a'\in U(a)} x_{a'}^k - 1 + \pi(\boldsymbol{a})\right\|_2^2\\
&= \frac{\rho_1}{2}\sum_{a\in A^{run}}\sum_{k\in K_1\cup K_2}\left\|x_a^k + \sum_{a'\in U(a)} x_{a'}^k + \sum_{k'\in K_1\cup K_2/\{k\}}\sum_{a\in A_{k'}} x_a^{k'} - 1 + \pi(\boldsymbol{a})\right\|_2^2\\
&= \frac{\rho_1}{2}\sum_{a\in A^{run}}\sum_{k\in K_1\cup K_2}\left\|\sum_{a'\in U(a)} x_{a'}^k + \varphi_a^k - 1 + \pi(\boldsymbol{a})\right\|_2^2
\end{aligned} \tag{17}$$

Furthermore, for any train k at most one arc can be selected from the set of incompatible arcs $\boldsymbol{U}(\boldsymbol{a})$, that is, the value of φ_a^k is equal to 1 or 0, where a is an element in the set of run arcs.

To eliminate the quadratic term when no conflict exists, we define the slack variable $\pi(\boldsymbol{a})$ such that it absorbs any 'unused' capacity in the incompatibility constraint:

$$\pi(\boldsymbol{a}) = 1 - \varphi_a^k - \sum_{a'\in U(a)} x_{a'}^k \tag{18}$$

Two cases arise:

Case 1: $\varphi_a^k = 0$

Then, .$\pi(\boldsymbol{a}) = 1 - \sum_{a'\in U(a)} x_{a'}^k$

Substituting this into the quadratic term yields zero, reducing the augmented LR function to the standard LR form Eq. (11).

Case 2: $\varphi_a^k \geq 1$:

Here $\pi(\boldsymbol{a})$ must be zero (since $\pi(\boldsymbol{a})\in[0,1]$, and $\varphi_a^k \geq 1$ implies $\pi(\boldsymbol{a}) \leq 0$.

Since $\left(x_a^k\right)^2 = x_a^k$ for binary variables, we can linearise the quadratic term as:

$$\left\| \sum_{a' \in U(a)} x_{a'}^k + \varphi_a^k - 1 \right\|_2^2 = \left(2 \times \varphi_a^k - 1\right) \times \sum_{a' \in U(a)} x_{a'}^k + \left(\varphi_a^k - 1\right)^2$$

The term $\left(\varphi_a^k - 1\right)^2$ is constant.

By considering these cases, the augmented LR function decomposes into a set of train-specific subproblems with linear objective functions:

$$Z_k^{ADMM} = \omega_1 \times \sum_{a \in A_k} c_a^k \times x_a^k + \omega_2 \times \sum_{a \in A_k} \tilde{c}_a^k \times x_a^k + \sum_{a \in A^{run}} x_a^k \times \sum_{k' \in K_1 \cup K_2} \lambda_a^{k'} + \sum_{a \in A^{run}} x_a^k \times \sum_{a' \in U(a)} \lambda_{a'}^k + \sum_{i \in V} \sum_{t \in T} \gamma_i^t \times \left(\sum_{\tau \in T} \sum_{a'' \in A^{dwell} \cup A^{stop}} x_{a''}^k + \sum_{a \in A^{run}} x_a^k \right) + \frac{\rho_1}{2} \times \sum_{a \in A^{run}} \left(2\varphi_a^k - 1\right) \times x_a^k + \frac{\rho_2}{2} \times \sum_{i \in V} \sum_{t \in T} \left(2\zeta_{i,t}^k - 2C_i + 1\right) \times x_a^k = \hat{c}_a^k \times x_a^k \tag{19}$$

We denote $\hat{c}_a^k$ as the modified arc cost for the augmented LR subproblem. Unlike in standard LR, $\hat{c}_a^k$ depends not only on the Lagrangian multipliers but also on the penalty parameters ρ_1 and ρ_2. Its form is:

$$\hat{c}_a^k = \begin{cases} \omega_1 \times c_a^k + \omega_2 \times \tilde{c}_a^k \, if a \notin A^{run} \cup A^{dwell} \cup A^{stop} and i \notin V \\ \omega_1 \times c_a^k + \omega_2 \times \tilde{c}_a^k + \sum_{k' \in K_1 \cup K_2} \lambda_a^{k'} + \sum_{a' \in U(a)} \lambda_{a'}^k + \frac{\rho_1}{2} \times \left(2\varphi_a^k - 1\right) if a \in A^{run} and i \notin V \\ \omega_1 \times c_a^k + \omega_2 \times \tilde{c}_a^k + \gamma_i^t + \frac{\rho_2}{2} \times \left(2\zeta_{i,t}^k - 2C_i + 1\right) if a \in A^{dwell} \cup A^{stop} and i \in V \\ \omega_1 \times c_a^k + \omega_2 \times \tilde{c}_a^k + \sum_{k' \in K_1 \cup K_2} \lambda_a^{k'} + \sum_{a' \in U(a)} \lambda_{a'}^k + \frac{\rho_1}{2} \times \left(2\varphi_a^k - 1\right) + \gamma_i^t \\ \quad + \frac{\rho_2}{2} \times \left(2\zeta_{i,t}^k - 2C_i + 1\right) if a \in A^{run} and i \in V \end{cases} \tag{20}$$

(4) ADMM iteration

After relaxation, augmentation, and linearisation, the original SETP model is decomposed into independent train-specific subproblems. The objective function for each train k is given by Eq. (19), with the modified arc cost $\hat{c}_a^k$(Eq. (20)). The ADMM framework alternates between (i) solving the single-train sub-problems and (ii) updating multipliers to enforce consensus constraints. The ADMM iteration proceeds as follows.

5.1. Subproblems solutions

For each train k, solve the train-specific subproblem defined by Eqs. (1), (3), (5) and (19) with the time-dependent shortest-path algorithm (see Algorithm 2 in Appendix B). This yields an updated selection of arcs x_a^k. Then update the general arc costs $\hat{c}_a^k$ using the latest solution for train k and the previous solutions of other trains. Note that the departure time window of extra trains and maximum departure shift of existing trains are addressed in Algorithm 2.

5.2. Multiplier update

Update the Lagrangian multipliers based on the penalty parameters ρ_1 and ρ_2. The updates for λ_a^k and γ_i^t at iteration $q+1$ are given by:

$$\lambda_a^k(q+1) = max\{0, \lambda_a^k(q) + \rho_1(q) \left(\sum_{k' \in K_1 \cup K_2} \sum_{a \in A_{k'}} x_a^{k'} + \sum_{a' \in U(a)} x_{a'}^k - 1 \right) \tag{21}$$

$$\gamma_i^t(q+1) = max\{0, \gamma_i^t(q) + \rho_2(q) \left(\sum_{k \in K_1 \cup K_2} \sum_{\tau \in T} \sum_{a'' \in A^{dwell} \cup A^{stop}} x_{a''}^k + \sum_{k \in K_1 \cup K_2} \sum_{a \in A^{run}} x_a^k - C_i \right) \tag{22}$$

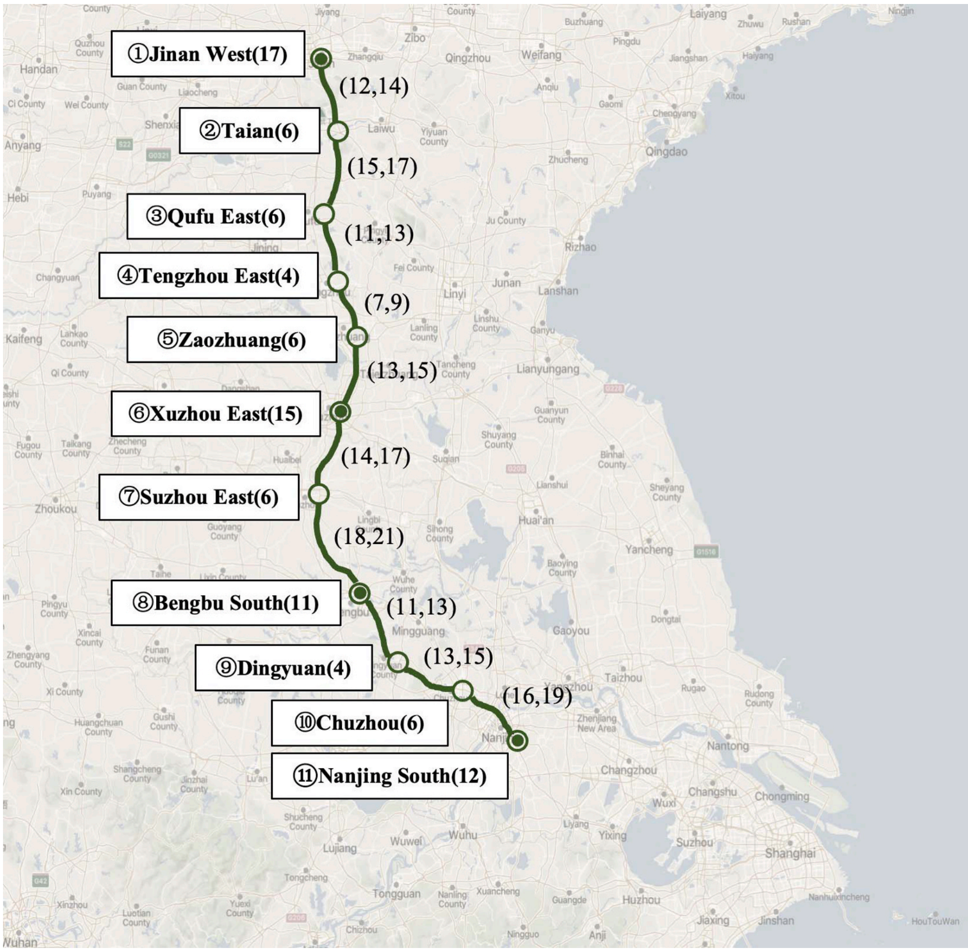

Fig. 6. Beijing-Shanghai high-speed railway test corridor (599 km, 11 stations). Stations represented by circles: filled circle within ring = major stations; empty circle = intermediate stations. Left panel shows station names with platform track capacities in parentheses. Numbers along track sections in brackets indicate running times (minutes) for G-trains (300 km/h) and D-trains (250 km/h) respectively.

5.3. Feasibility check and penalty update

Compute the primal residual ***res*** (defined in Eq. (23)) to measure violations, then update ρ_1 and ρ_2 according to Eq. (24).

$$res = max\left\{\sum_{a\in A^{run}}\sum_{k\in K_1\cup K_2}\left(\sum_{k'\in K_1\cup K_2}\sum_{a\in A_{k'}}x_a^{k'}+\sum_{a'\in U(a)}x_{a'}^{k}-1\right)+\right.$$
$$\left.+\sum_{i\in V}\sum_{t\in T}\left(\sum_{k\in K_1\cup K_2}\sum_{\tau\in T}\sum_{a''\in A^{dwell}\cup A^{stop}}x_{a''}^{k}+\sum_{k\in K_1\cup K_2}\sum_{a\in A^{run}}x_a^k-C_i\right)\right. \tag{23}$$

We then update the penalty parameters ρ_1 and ρ_2 based on the feasibility of the current solution. For each penalty parameter $\rho\in\{\rho_1,\rho_2\}$ at iteration q, we adopt the following update rule:

$$\rho(q+1)=\begin{cases}\rho(q)\,if\,|res^q|\le\theta|res^{q-1}|\\ \beta\times\rho(q)\,if\,|res^q|>\theta|res^{q-1}|\end{cases} \tag{24}$$

Here, $\beta>1$ and $0<\theta<1$. The values of β and θ are chosen based on previous studies (Xiu et al., 2024; Zhan et al., 2022b; Zhang et al., 2019).

The ADMM iterative procedure stops when the iteration count Q is reached. The iterative ADMM procedure used to solve the decomposed SETP model is given in Algorithm 1.

Algorithm 1. ADMM-based decomposition for solving SETP

Input:
$G=(N,A)$: Space-time network and arc costs
K_1 : Set of existing trains (with known schedules)
K_2 : Set of extra trains (with given time windows, stop patterns, and section running times)
Q: Maximum iteration step
Step 1: Initialisation
Set iteration counter $q\leftarrow 0$

(continued on next page)

(*continued*)

Algorithm 1. ADMM-based decomposition for solving SETP

Initialise the penalty parameters $\rho_1(0), \rho_2(0)$
For each train $k \in K_1 \cup K_2$:
  Initialise scheduling variables $x_a^k = 0$ for all arcs $a \in A$
  Initialise Lagrangian multipliers $\lambda_a^k(0) = 0, \gamma_i^t(0) = 0$
Step 2: Initial scheduling of all trains
For each train $k \in K_1 \cup K_2$:
  Update arc costs according to Eq. (20);
  With other trains fixed, apply Algorithm 2 to obtain train k's shortest path. Let $x_a^k = 1$ for each arc a on this shortest path.
Step 3: ADMM iterative procedure
While $q < Q$:
  For each train $k \in K_1 \cup K_2$:
   Reset $x_a^k = 0$ for all arcs $a \in A_k$
   Solve the shortest path problem again for train k using the time-dependent shortest-path algorithm (Algorithm 2 in Appendix B).
   Assign $x_a^k = 1$ along the obtained shortest path
   Update the general arc costs by Eq. (20) based on the latest solution for train k and previous solutions of other trains.
  End for
  Update Lagrangian multipliers λ_a^k, γ_i^t using the current solution and the update rules Eq. (21) and Eq. (22)
  Compute the primal residuals and update the penalty parameters ρ_1 and ρ_2 using Eq. (23) and Eq. (24)
  Increment the iteration counter $q \leftarrow q + 1$
Step 4: Output
  Output the final x_a^k for all scheduled trains

Convergence of multi-block ADMM applied to integer programmes is not theoretically guaranteed (Yao et al., 2019). The penalty parameter ρ affects both convergence speed and solution quality: larger values accelerate feasibility convergence but may yield inferior objectives, while smaller values achieve better solutions at the cost of more iterations (Gao and Niu, 2021; Xiu et al., 2024). We adopt dynamic parameter updates (Eq. (24) and evaluate solution quality through comparisons with LR solutions (Section 6.6). Sensitivity analysis of ρ is provided in Section 6.5.

Table 2
Default parameter values for computational experiments.

Train operation parameters	Values
h^{arr}	3 min
h^{dep}	3 min
$tw_{k,i}^{min}$	2 min
$tw_{k,i}^{max}$	10 min for existing trains; 20 min for extra trains
Δt_{stop}	1 min
ADMM parameters	
Q	25
$\lambda_a^k(0), \gamma_i^t(0)$	0
$\rho_1(0), \rho_2(0)$	2
β	1.5
θ	0.5

6. Experimental results

6.1. Overview of instances and parameter settings

This section describes the computational instances and parameter configurations used in our experiments. The test instances are derived from the Beijing–Shanghai high-speed railway corridor in China — one of the country's most critical and heavily utilised lines. As illustrated in Fig. 6, this double-track corridor spans approximately 599 km and includes 11 stations, where major stations (denoted by double circles) serve as origins or termini. The station-specific track capacities for the downward direction are indicated in parentheses following each station's name.

A baseline timetable is obtained from a representative non-holiday weekday in December 2019 (see Fig. 7), with all trains on this corridor being high-speed "G" category trains running at 300 km/h. The travel times for both high-speed and slower trains are indicated alongside the corridor in Fig. 6. The planning horizon covers the period from 6:00 to 18:00, encompassing a total of 70 high-speed train services: 60 from Jinan West to Nanjing South, three from Xuzhou East to Nanjing

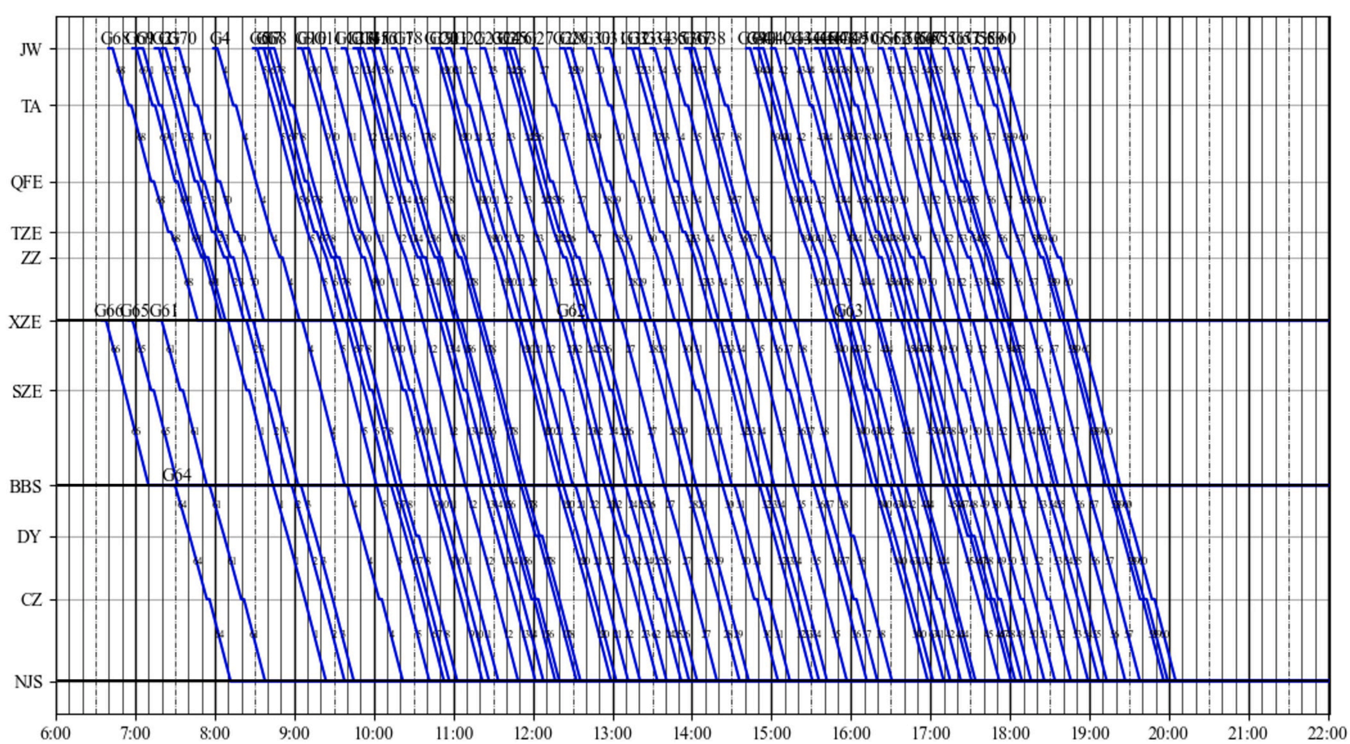

Fig. 7. Baseline timetable: 70 existing G-trains, 12-hour horizon (6:00–18:00). Real data from December 2019 (non-holiday weekday). Vertical: 11 stations (Jinan West to Nanjing South); horizontal: time.

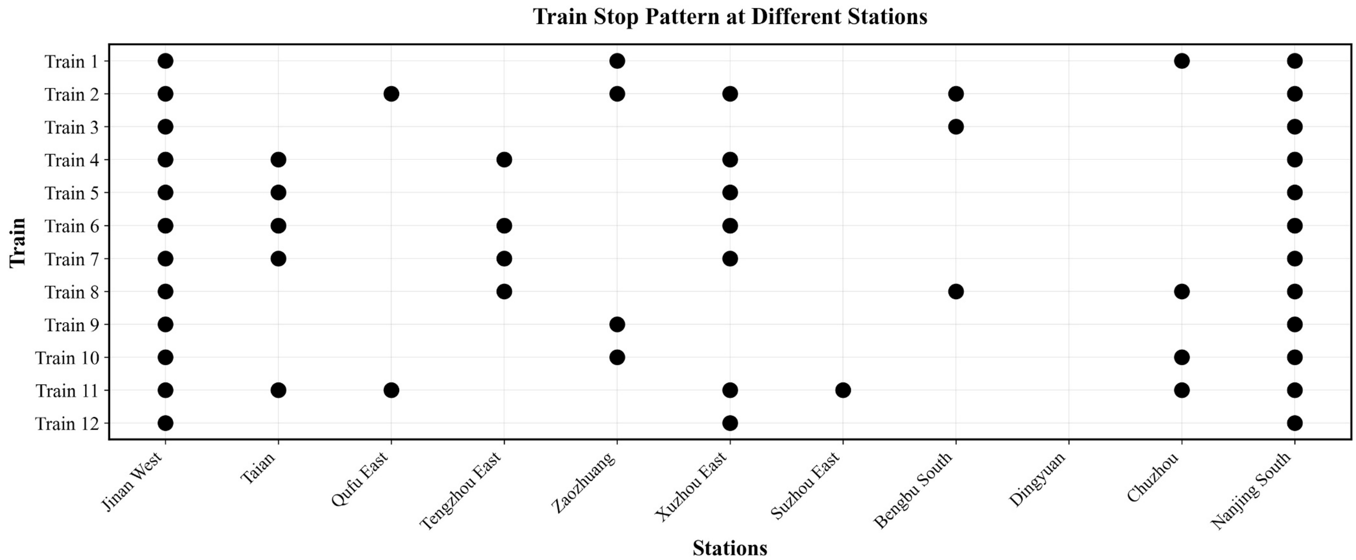

Fig. 8. Stopping patterns of 12 extra trains. Trains 1–12 (vertical) versus stations (horizontal). Black dots = mandatory stops. All trains run from Jinan West to Nanjing South with departure time windows.

Table 3
Outcomes of train insertion 12 mix-speed trains with varying weight ratios and departure time windows. In the Experiment Settings column, the tuple denotes: maximum departure shifts for existing trains (minutes), half-width of departure window for extra trains (minutes), number of extra trains to be inserted.

$\omega_1 : \omega_2$	Experiment Settings	Deviation Time(min)	Travel Time(min)	Unplanned Stops		Iteration		Computation Time (s)
				Existing	Extra	Best	Feasible	
1:1	(5,15,12)	77	1944	11	6	14	13	779
	(5,30,12)	93	1893	10	6	11	10	692
	(5,45,12)	113	1850	12	1	10	9	614
2:1	(5,15,12)	71	1953	10	7	15	13	909
	(5,30,12)	70	1894	7	2	13	12	796
	(5,45,12)	78	1868	9	1	13	12	768
4:1	(5,15,12)	63	1959	11	10	17	16	792
	(5,30,12)	70	1894	7	2	17	16	1015
	(5,45,12)	59	1880	9	4	15	14	895
10:1	(5,15,12)	59	1979	9	8	19	18	1061
	(5,30,12)	70	1894	7	2	18	17	845
	(5,45,12)	59	1883	9	1	19	18	1133

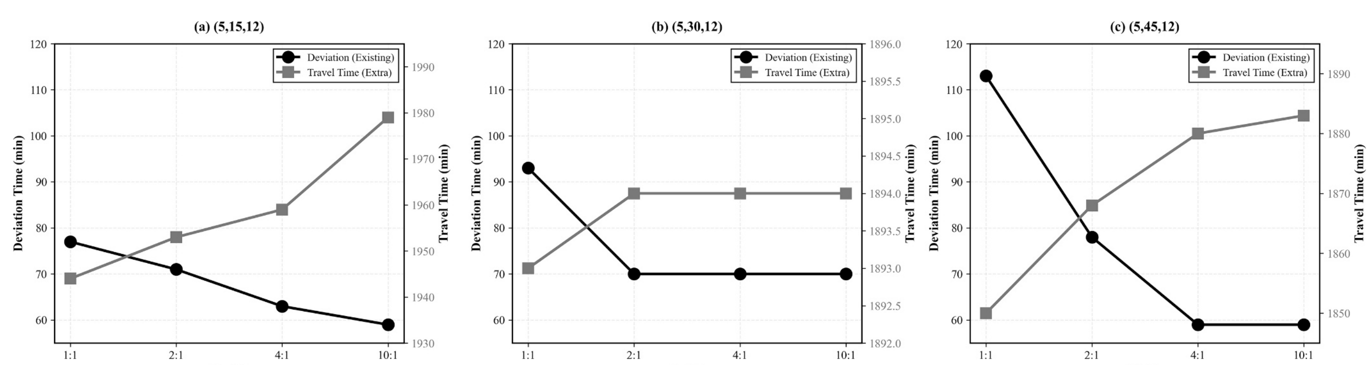

Fig. 9. Trade-offs between existing train deviations and extra train travel times. Inserting 12 mixed-speed trains with departure flexibility: (a) ± 15 min, (b) ± 30 min, (c) ± 45 min. Left Y-axis: existing train deviations (min); right Y-axis: extra train travel times (min); X-axis: weight ratios $\omega_1 : \omega_2$(1:1 to 10:1). Higher ratios prioritize existing train punctuality.

South, one from Bengbu South to Nanjing South, three from Xuzhou East to Bengbu South, and three linking Jinan West and Xuzhou East. Two categories of extra trains are inserted: fast G (300 km/h) and slower D (250 km/h).

Table 2 lists the default model parameter values. The minimum headway values h^{arr} and $\mathbf{h}^{\mathbf{dep}}$ are both set to 3 min, which have been widely adopted in the literature for the Beijing-Shanghai railway (e.g., Zhang et al., 2019; Zhan et al., 2022a; Gao and Niu, 2021). The minimum dwell time of 2 min allows sufficient time for passenger boarding/alighting, while the maximum dwell times (10 min for existing trains, 20 min for extra trains) prevent excessive waiting while maintaining flexibility. In addition, the extra trains are assigned specified departure

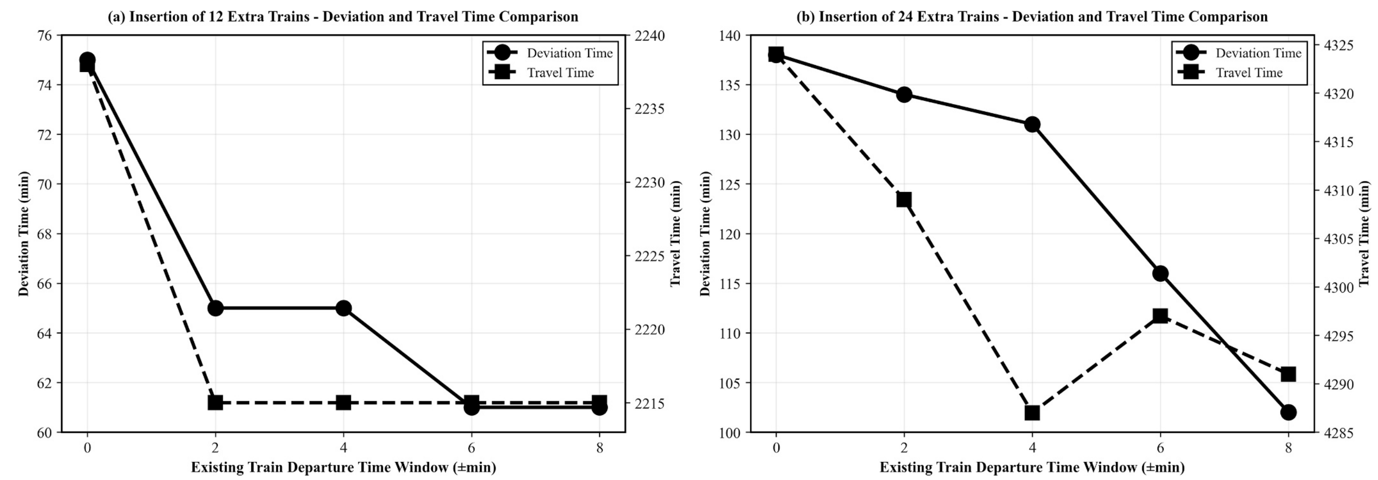

Fig. 10. Impact of existing trains' departure time shifts. Inserting D-trains (250 km/h): (a) 12 trains, $\omega_1 : \omega_2 = 4 : 1$; (b) 24 trains, $\omega_1 : \omega_2 = 2 : 1$. Extra train window: ±15 min. X-axis: existing train shift (0 to ± 8 min). Left Y-axis: existing train deviations (min); right Y-axis: extra train travel times (min).

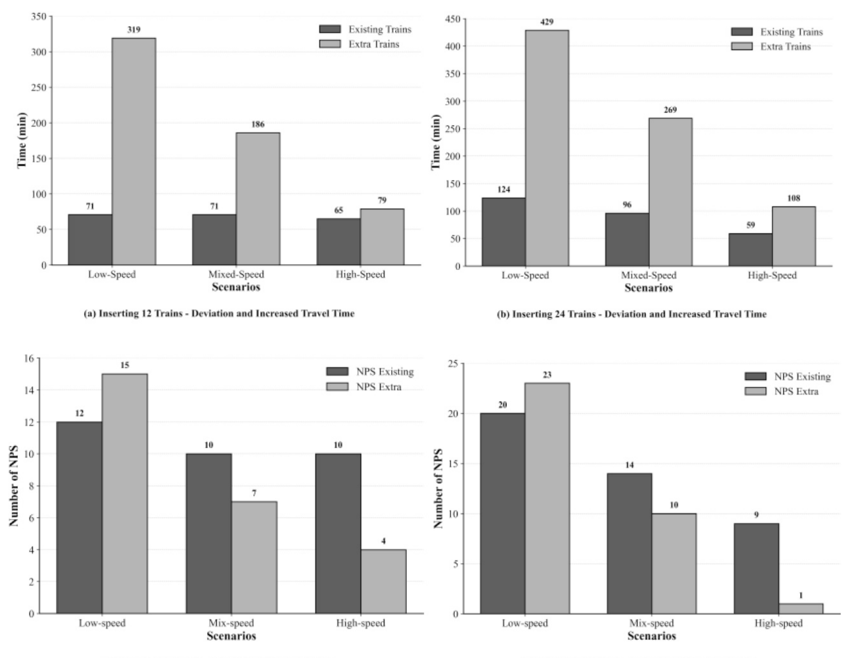

Fig. 11. Performance comparison of inserting different train speed classes. Three configurations tested: high-speed only, mixed-speed, and low-speed only. (a) 12 trains—existing trains' deviation and extra trains' increased travel times; (b) 24 trains—existing trains' deviation and extra trains' increased travel times; (c) 12 trains—non-planned stops (NPS); (d) 24 trains—NPS. Dark bars: existing trains; light bars: extra trains.

time windows ($\boldsymbol{DT_k}^e, \boldsymbol{DT_k}^l$), while existing trains are provided with a maximum departure time shift $\boldsymbol{sh_k}$ at origin stations. Departure time windows are operator-specified inputs, typically determined by passenger flow analysis, rolling stock availability, and commercial requirements (Gao et al., 2018; Gao and Niu, 2021). Following established SETP practice, we conduct sensitivity analyses to demonstrate how these parameters affect scheduling outcomes. Details regarding these inputs, as well as weighting factors $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ will be discussed in subsequent experimental sections. Furthermore, we set the artificial arc cost $\boldsymbol{R_k^{existing}}$ as 1000 to prevent existing trains from being unscheduled, while $\boldsymbol{R_k^{extra}}$ is set to 180 min for G-trains and 200 min for D-trains. The artificial arc cost $\boldsymbol{R_k^{extra}}$ represents the maximum acceptable travel time, set as approximately 1.5 times the minimum travel time for each train type. All computations were performed on a computer equipped with an Intel(R) Core (TM) i5-12490F CPU at 3.0 GHz and 32 GB of RAM. All programming was implemented using Python 3.7 on a Microsoft Windows 11 (64-bit) operating system.

6.2. Impact of departure–window and weight ratio $\omega_1 : \omega_2$

This section evaluates how the weight ratio ($\omega_1 : \boldsymbol{\omega}_2$) and the departure-window width of extra trains affect scheduling performance. The analysis tracks four metrics: deviation of existing trains, travel time of extra trains, number of unplanned stops, and computational performance.

In this experiment, 12 mixed-speed trains (6 high-speed and 6 low-speed) are integrated into the existing timetable. All extra trains depart from Jinan West and terminate at Nanjing South, with their stopping patterns depicted in Fig. 8. Scheduled stops are represented by solid circles, each requiring a minimum dwell time of 2 min. Weight ratios $\omega_1 : \boldsymbol{\omega}_2$ range from 1:1 to 10:1, while departure-time flexibility

Table 4
Results with different quadratic penalty parameters.

Experiment Settings	ρ	Z_{fea}	Z_{best}	$Gap_{improve}$(%)	Iteration	
					Feasible	Best
(4,15,12)	1	2304	2236	2.95	15	16
	2	2245	2141	4.63	16	17
	4	2425	2269	6.43	12	13
	8	2375	2251	5.22	11	12
(6,30,24)	1	4089	3989	2.45	15	16
	2	4089	3977	2.74	13	14
	4	4075	3991	2.06	12	13
	8	4007	3929	1.95	10	11

Note: Z_{fea} is the first feasible solution, Z_{best} is the final solution. The gap measures the improvement from first feasible to final solution: $Gap_{improve}$= (Z_{fea}-Z_{best})/ Z_{fea} $\times$ 100%.

takes ± 15, ±30, and ± 45 min. For example, under ± 15-minute flexibility, the first extra train's departure window is [6:15, 6:45]. Expanding flexibility to ± 30 min adjusts this to [6:00, 7:00], while ± 45-minute flexibility extends it to [6:00, 7:15]. Due to the planning horizon restriction of [6:00, 18:00], the start time for the ± 45-minute case is fixed at 6:00 instead of 5:45. Subsequent trains' departure windows shift forward by 1-hour increments; for instance, under ± 45-minute flexibility, the second train's window becomes [6:45, 8:15].

Table 3 summarises the results, with the Experiment Settings column detailing the departure time windows for extra trains, allowable departure shifts at origin station for existing trains (±5 minutes) as well as the number of extra trains inserted. Wider windows shorten extra trains' travel time and barely affect existing trains _ except at the 1:1 ratio, where deviation grows. At a weight ratio of 1:1, extending flexibility from ± 15 to ± 45 min reduces extra train travel times from 1944 to 1850 min but increases deviation times for existing trains from 77 to 113 min. This trade-off reflects greater path allocation freedom for extra trains at the expense of existing train punctuality. At higher weight ratios, the effect of extra trains' flexibility on existing train deviation times diminishes. Deviation times stabilise across all flexibility levels due to stricter prioritisation of existing train punctuality. However, extra trains still benefit from increased flexibility; for example, at a 10:1 wt ratio, their travel times decrease from 1979 min (±15) to 1883 min (±45). Trends in non-scheduled stops align with travel time and deviation patterns. The influence mechanism differs by train type. For extra trains, each unplanned stop uses an extra dwell arc with cost Δt_{stop}, directly increasing travel time. For existing trains, extra dwell arcs have zero deviation cost, but they delay subsequent arrivals at downstream stations; since deviation cost is calculated as the difference between actual and scheduled arrival times, these delays propagate and increase total deviation cost. Thus, unplanned stops serve as scheduling buffers to resolve conflicts, regulated by the weight ratio $\omega_1 : \omega_2$. At weight ratios of 1:1, wider departure windows reduce extra train stops from 11 (±15 min) to 9 (±45 min) while slightly increasing unplanned stops for existing trains. At higher weight ratios, unplanned stops for existing trains remain stable due to stricter punctuality constraints, and extra train stops stay consistently low regardless of flexibility levels.

Fig. 9 illustrates the trade-offs between existing train deviation times and extra train travel times across different weight ratios when departure time flexibilities are fixed. Across all departure time window settings (±15, ±30, and ± 45 min), existing train deviation times decrease as the weight ratio $\omega_1 : \omega_2$ increases, while extra train travel times increase. Additionally, the average deviation time for existing trains, calculated as the mean across all time window settings, drops from 94.3 min at 1:1 to 62.7 min at 10:1, reflecting improved punctuality as existing trains are prioritised. Conversely, the average travel time for extra trains rises from 1895.7 min to 1918.7 min, demonstrating the trade-off where stricter prioritisation of existing trains leads to a moderate decline in extra train efficiency. Through Table 3 we find that increasing the weight ratio $\omega_1 : \omega$ reduces unplanned stops for existing trains. At 1:1, existing trains make an average of 11.3 stops across all departure time window settings, decreasing to 7.7 stops at 10:1 due to prioritising punctuality. For extra trains, unplanned stops decrease slightly from 10.3 to 8.7 stops. This suggests that the departure time flexibility for extra trains may help mitigate the impact of higher weight ratios on their performance.

Moreover, the results in Table 3 show that the ADMM-based algorithm converges efficiently within 25 iterations, with computation times ranging from 614 to 1133 s. Higher weight ratios $\omega_1 : \omega_2$, which emphasise existing train punctuality, increase computation time due to stricter constraints that reduce flexibility and complicate scheduling. Additionally, wider departure time flexibility does not always increase computation times; rather, it generally reduces them by improving solution feasibility and thus reducing the number of iterations needed to obtain feasible and best solutions.

6.3. *The effect of shift to existing train's departure time*

In this experiment, we investigate the impact of the allowable departure time shift of 70 existing trains while integrating 12 and 24 slow extra trains into a high-speed rail timetable. The extra trains' departure time window is set at ± 15 min. For the 24-train scenario, two extra trains per hour are inserted with identical stop patterns, split evenly

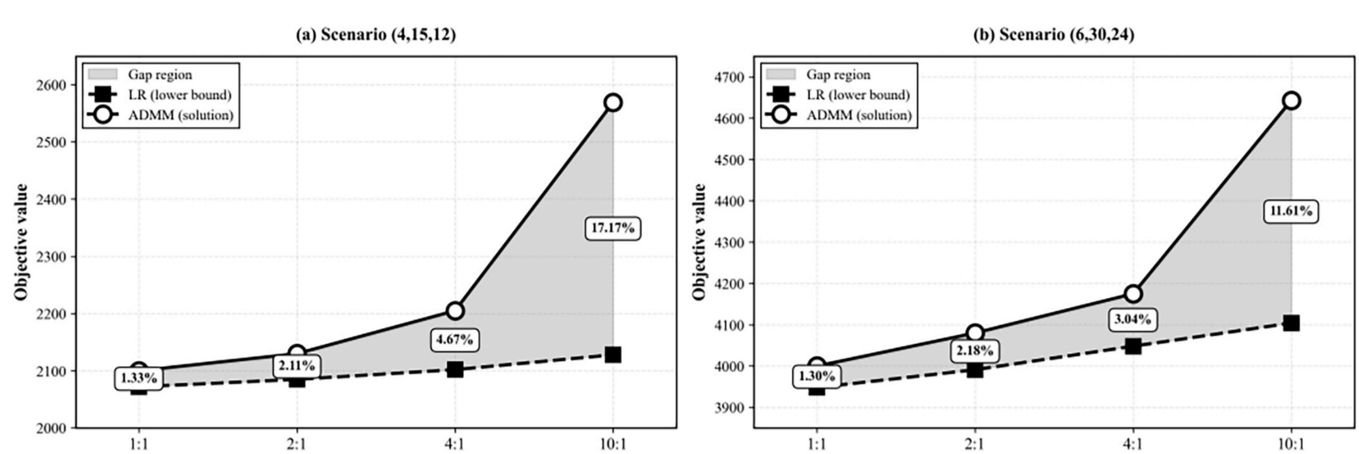

Fig. 12. ADMM solutions versus Lagrangian Relaxation lower bounds. Two scenarios: (a) (4,15,12); (b) (6,30,24). X-axis: weight ratios $\omega 1 : \omega 2$ (1:1 to 10:1); Y-axis: objective value (min). Circles = LR lower bound; squares = ADMM solution. Numbers = optimality gaps (%). Gaps range from 1.3% at 1:1 to 17% at 10:1, average 5.43%.

between two sub-windows, such as 7:00–7:30 and 7:30–8:00. The scheduling algorithm applies a weight ratio of 4:1 for inserting 12 trains and 2:1 for inserting 24 trains, considering the higher complexity of accommodating a larger number of extra trains. In these experiments, all extra trains are inserted.

Fig. 10 shows that increasing departure time flexibility for existing trains reduces their deviation times and improves the travel times of extra trains. For the scenario of inserting 12 extra trains, deviation times drop from 75 min at 0 min to 61 min at $\pm$ 6 min, while extra trains' travel times decrease and stabilise after $\pm$ 2 min. In the 24-train scenario, deviation times decline more notably, from 138 min at 0 min to 102 min at $\pm$ 8 min, with extra train travel times decreasing from 4324 to 4291 min, showing minor fluctuations between $\pm$ 4 and $\pm$ 6 min. These results indicate that moderately increasing flexibility is beneficial in congested scenarios with more trains, where more trains lead to increased scheduling conflicts. However, it is important to note that excessively increasing the existing trains' maximum shift may not necessarily yield additional benefits, as observed when inserting 12 extra trains.

Combining these findings with the previous experiments, it becomes evident that selecting appropriate departure time windows for extra trains, shift values for existing trains, and weight ratios is critical for balancing solution quality and computational efficiency.

6.4. The effect of the extra train types

This experiment evaluates the effects of inserting different types of extra trains into an existing high-speed railway timetable. The inserted trains are categorised into three groups: (i) 12 or 24 high-speed trains, (ii) a mix of 12 or 24 high-speed and low-speed trains, and (iii) 12 or 24 low-speed trains, with a weight ratio of 2:1. Departure time shifts are set at $\pm$ 4 min for the 12-train scenario and $\pm$ 6 min for the 24-train scenario for existing trains, while extra trains are assigned a broader departure window of $\pm$ 15 min. The impact is assessed using increased travel time of extra trains, deviation time for existing trains, and non-planned stops (NPS) for both existing and extra trains. Increased travel time is used instead of total travel time to normalise the inherent differences between speed groups.

Fig. 11 shows the impact of inserting different types of extra trains into a high-speed train timetable, evaluating deviation times for existing trains (Fig. 11(a)), total increased travel times for extra trains (Fig. 11(b)), and extra stops for both types (Fig. 11(c) and Fig. 11(d)). When inserting extra trains, high-speed trains perform best, with minimal disruptions to the timetable. For example, in the 12-train scenario, deviation times for existing trains are reduced to 65 min, while extra train travel time increases by only 79 min. These values further improve in the 24-train scenario, with deviation times dropping to 59 min and increased travel time to 108 min. In contrast, low-speed trains cause significant conflict, with deviation times reaching 124 min and increased travel time increasing to 429 min in the 24-train scenario. Mixed-speed trains offer moderate improvements, reducing these values to 96 and 269 min, respectively, but still introduce inefficiencies compared to inserting high-speed trains. The results indicate that homogeneous high-speed train insertion ensures smoother integration and maintains operational efficiency, while mixed and low-speed trains disrupt the timetable and need more modifications to the existing timetable. While the model does not explicitly constrain the number of overtaking operations, overtaking is implicitly discouraged by increasing the travel time.

6.5. Sensitivity analysis of quadratic penalty parameters

This subsection conducts a sensitivity test on the ADMM quadratic-penalty ρ_1 and ρ_2. To assess their influence, we set $\rho = \rho_1(0) = \rho_2(0)$ and test a range of initial penalty values {1, 2, 4, 8}, with the weight ratio fixed at 2:1. The experiments involved inserting mixed-speed trains in two scenarios: (4,15,12) and (6,30,24), with the results summarised in Table 4. The findings suggest that larger penalty values ρ generally facilitate earlier convergence to a feasible solution. However, smaller penalty values tend to achieve lower Z_{best}. Specifically, in scenario (4,15,12), $\rho = 2$ achieves the smallest Z_{best}, while in scenario (6,30,24), $\rho = 4$ provides the best results in terms of solution quality. These findings suggest a trade-off between runtime and solution quality when tuning ρ.

6.6. Comparison with commercial solver and Lagrangian Relaxation method

To evaluate the solution quality of the ADMM algorithm, we compare the results obtained by our ADMM algorithm with Gurobi and employ the Lagrangian Relaxation method to establish lower bounds.

We construct three instances from the Beijing–Shanghai corridor. All instances share the same headway and maximum/minimum dwell times, with departure time windows ($\pm$4 min for existing trains, $\pm$5 min for extra trains) and weight ratio (4:1). Gurobi is configured with MIPGap = 0.5% and TimeLimit = 3600 s. On the small-scale instance (3 existing + 2 slow extra trains, 331,437 binary variables), Gurobi finds the globally optimal solution with objective 322 in less than 1 s—identical to the ADMM result (Gap = 0%). On a medium instance (14 existing + 6 slow extra trains, 1,334,262 binary variables), Gurobi achieves the optimal objective 664 in 464.45 s, while ADMM achieves 667, yielding a gap of 0.45% relative to the verified optimum. For the full-scale instance (82 trains, 5,387,342 binary variables), Gurobi does not converge to an optimal solution within the imposed time limit of 3600 s, indicating the computational intractability of solving the monolithic ILP directly at this scale. In contrast, ADMM produces a feasible solution with zero constraint violations within 15 min.

For large-scale instances where Gurobi is intractable, we employ the LR method to establish lower bounds. The LR method relaxes the same coupling constraints (4) and (7) as ADMM but uses only linear penalty terms without quadratic augmentation. Importantly, the LR relaxation provides a lower bound on the optimal objective value since it relaxes capacity and headway constraints. The LR solution may violate these constraints and is generally infeasible for actual operations, whereas ADMM explicitly enforces all constraints through iterative coordination, guaranteeing feasibility. The detailed LR solution algorithm is provided in Appendix D. We select two scenarios for comparison: (4,15,12) and (6,30,24), each involving the insertion of 12 and 24 mixed-speed trains. For each scenario, we compare the objective function values obtained by ADMM and LR under different weight ratios $\omega_1 : \omega_2$. Let Z^{ADMM}_{best} denote the best feasible objective value found by the ADMM algorithm, and Z^{LR}_{best} denote the best lower bound obtained by the LR algorithm over 100 iterations. The optimality gap is defined as $Gap_{opt} = \left(Z^{ADMM}_{best} - Z^{LR}_{best}\right)/Z^{ADMM}_{best} \times 100\%$, which measures how close the ADMM solution is to the LR lower bound.

The results shown in Fig. 12 reveal several important observations. First, when prioritising both objectives equally (1:1 ratio), ADMM achieves near-optimal solutions with gaps of only 1.33% and 1.30%, demonstrating good solution quality. Second, as the weight ratio increases to 10:1 (strongly prioritising existing train punctuality), gaps expand substantially to 17.17% (scenario 1) and 11.61% (scenario 2). This trend occurs because higher weight ratios create tighter constraints on existing train movements, reducing solution space flexibility and making the problem more challenging for decomposition methods. The decomposition may require additional iterations to fully coordinate train movements under stricter punctuality requirements.

Third, scenario (6,30,24) consistently shows smaller gaps than scenario (4,15,12) at higher weight ratios (e.g., 11.61% vs. 17.17% at 10:1). This suggests that wider departure time windows provide more scheduling flexibility, enabling ADMM to find better solutions even with more trains (24 vs. 12). Finally, across all eight test points (four weight ratios × two scenarios), the average gap is 5.43%, which is comparable to gaps reported for LR-based heuristics in similar train scheduling problems (Meng and Zhou, 2014; Zhan et al., 2022a).

Importantly, ADMM achieves these gaps while guaranteeing feasibility, whereas LR solutions typically require additional repair procedures to restore constraint satisfaction. Moreover, ADMM typically obtains solutions within 16–19 iterations (see Table 3), demonstrating that ADMM provides a favourable balance between solution quality and computational tractability for practical railway operations.

7. Conclusions

This study tackles the problem of scheduling extra train paths (SETP) within an existing timetable on a double-track high-speed railway corridor. The SETP problem is inherently complex due to its combinatorial nature and the conflicting objectives of minimising extra train travel times while preserving the punctuality of existing services.

We formulate SETP as an ILP problem on a space–time network and introduce an ADMM-based decomposition approach for the first time in this context. The approach decomposes the ILP into independent, train-specific subproblems that are solved via a time-dependent shortest-path algorithm. The ADMM iterations alternate between solving these subproblems and updating the Lagrangian multipliers as well as ADMM penalties, significantly reducing computational effort.

On the Beijing–Shanghai corridor with 70 existing trains and up to 24 extra trains, the algorithm converged within 25 iterations and 10–19 min of CPU time, producing high-quality conflict-free timetables in all experiments. By introducing varying degrees of flexibility, we find that allowing moderate departure time adjustments for existing trains at their origin stations effectively reduces both their deviations and the travel time of extra trains. Furthermore, widening the departure window for extra trains decreases their travel times, while having minimal impact on the baseline services—except when equal weight is assigned to both train types in the objective function. In this case, a slight increase in deviation for existing trains is observed. Finally, the inclusion of trains with different speed classes introduces additional scheduling complexity. Nevertheless, our proposed framework can resolve these conflicts and offer an adaptable tool for railway operators to respond flexibly to fluctuations in demand or unexpected events.

Currently, running times are fixed and do not explicitly capture acceleration/deceleration at stops. This could be addressed by extending the space–time network with four running arc types (Pass-Pass, Stop-Pass, Pass-Stop, Stop-Stop) following Jiang et al. (2017), with corresponding modifications to track stopping states in the shortest path algorithm. Additionally, the computational experiments are limited to a single corridor; generalisation to network-level problems requires further validation. Regarding scalability, ADMM decomposes the problem into train-specific subproblems solved via shortest path algorithms. The main computational factors for larger instances are the space–time network size and the number of iterations required for convergence. Potential enhancements include warm-starting multipliers from previous solutions and adaptive penalty parameter tuning (Boyd et al., 2011). Future work could integrate rolling-stock circulations into SETP or develop more detailed models including platform assignment to increase practical applicability.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A:. Modified arc cost after Lagrangian Relaxation

$$Z_L^k = \omega_1 \times \sum\nolimits_{a\in A_k} c_a^k \times x_a^k + \omega_2 \times \sum\nolimits_{a\in A_k} \widetilde{c}_a^k \times x_a^k + \sum_{a\in A^{run}} x_a^k \times \sum_{k'\in K_1\cup K_2} \lambda_a^{k'} + \sum_{a\in A^{run}} x_a^k \times \sum_{a'\in U(a)} \lambda_{a'}^k + \sum_{i\in V}\sum\nolimits_{t\in T} \gamma_i^t \times \left(\sum_{\tau\in T}\sum_{a''\in A^{dwell}\cup A^{stop}} x_{a''}^k + \sum_{a\in A^{run}} x_a^k\right) = \dot{c}_a^k \times x_a^k$$

$$\dot{c}_a^k = \begin{cases} \omega_1 \times c_a^k + \omega_2 \times \widetilde{c}_a^k \, if a \notin A^{run} \cup A^{dwell} \cup A^{stop} and i \notin V \\ \omega_1 \times c_a^k + \omega_2 \times \widetilde{c}_a^k + \sum\nolimits_{k'\in K_1\cup K_2} \lambda_a^{k'} + \sum\nolimits_{a'\in U(a)} \lambda_{a'}^k \, if a \in A^{run} and i \notin V \\ \omega_1 \times c_a^k + \omega_2 \times \widetilde{c}_a^k + \gamma_i^t if a \in A^{dwell} \cup A^{stop} and i \in V \\ \omega_1 \times c_a^k + \omega_2 \times \widetilde{c}_a^k + \sum\nolimits_{k'\in K_1\cup K_2} \lambda_a^{k'} + \sum\nolimits_{a'\in U(a)} \lambda_{a'}^k + \gamma_i^t if a \in A^{run} and i \in V (A.1) \end{cases}$$

Appendix B:. Time-dependent shortest path algorithm

Algorithm 2: Time-dependent shortest path algorithm for solving subproblems

Input:
A space–time network $G = (N, A)$;
Set of existing trains K_1 (with known schedules) with their scheduled departure times td_{k,O_k} at origin stations and maximum allowable time shift sh_k
Set of extra trains K_2 with given departure time windows $[DT_k{}^e, DT_k{}^l]$
For each train $k \in K_1 \cup K_2$, space–time network, origin o_k, destination d_k, earliest departure time e_k and latest arrival time l_k, arc costs c_a^k or $\widetilde{c}_a^k$
Coefficient weight ω_1 and ω_2
Step 1: Initialisation
For train $k = 1$ to $|K_1 \cup K_2|$
If $k \in K_1$
Define the feasible initial departure time range as $[td_{k,O_k} - sh_k, td_{k,O_k} + sh_k]$
Else if $k \in K_2$

(continued on next page)

(*continued*)

Algorithm 2: Time-dependent shortest path algorithm for solving subproblems

Define the feasible initial departure time range as $\left[\boldsymbol{DT_k}^{e}, \boldsymbol{DT_k}^{l}\right]$
For physical link $(\boldsymbol{i}, \boldsymbol{j}) = 1$ to $|\boldsymbol{E_k}|$
For each time $\boldsymbol{t} \in [\boldsymbol{e_k}, \boldsymbol{l_k}]$
If $\boldsymbol{i} = \boldsymbol{o_k}$ and $\boldsymbol{t}$ is in the feasible start time range:
$\delta_{i,t}^{k} = 0$ // label cost for $\boldsymbol{k}$ at node $(\boldsymbol{i}, \boldsymbol{t})$
Else:
$\delta_{i,t}^{k} = \infty$
End if
Set train node precedence $\boldsymbol{PN}_{i,t} = -1$, time precedence $\boldsymbol{PT}_{i,t} = -1$
End for // times
End for // links
End for // trains
Step 2: Forward finding the best node on the shortest path
For train $\boldsymbol{k} = 1$ to $|\boldsymbol{K}_1 \cup \boldsymbol{K}_2|$
For node $\boldsymbol{i} = 1$ to $|\boldsymbol{N_k}|$
For time $\boldsymbol{t} \in [\boldsymbol{e_k}, \boldsymbol{l_k}]$
If $\boldsymbol{i} = \boldsymbol{o_k}$
If $\boldsymbol{t} \notin \left[\boldsymbol{td}_{k,O_k} - \boldsymbol{sh_k}, \boldsymbol{td}_{k,O_k} + \boldsymbol{sh_k}\right]$ for $\boldsymbol{k} \in \boldsymbol{K}_1$ or $\boldsymbol{t} \notin \left[\boldsymbol{DT_k}^{e}, \boldsymbol{DT_k}^{l}\right]$ for $\boldsymbol{k} \in \boldsymbol{K}_2$
Continue // skip if outside departure time window
Derive feasible downstream node $\boldsymbol{j}$ for $\boldsymbol{i}$, feasible downstream time $\boldsymbol{t'}$ for $\boldsymbol{t}$ based on the stopping pattern of train $\boldsymbol{k}$;
Calculate the general cost $\widehat{\boldsymbol{c}}_a^k$ of arc $\boldsymbol{a}$ according to Eq. (20)
If $(\delta_{i,t}^{k} + \widehat{\boldsymbol{c}}_a^k < \delta_{j,t'}^{k})$
$\delta_{j,t'}^{k} = \delta_{i,t}^{k} + \widehat{\boldsymbol{c}}_a^k$
$\boldsymbol{PN}_{j,t'} = \boldsymbol{i}, \boldsymbol{PT}_{j,t'} = \boldsymbol{t}$
End if
End for $//\boldsymbol{t}$
End for $//\boldsymbol{i}$
End for $//\boldsymbol{k}$
Step 3: Backtrack the time-dependent shortest path
Find the last node $(\boldsymbol{d_k}, \boldsymbol{t}^*)$ of train k's shortest path
Backtrack from $(\overline{\boldsymbol{d}}_{\boldsymbol{k}}, \boldsymbol{t}^*)$ to $(\overline{\boldsymbol{o}}_{\boldsymbol{k}}, \boldsymbol{t}_0)$ according to $\boldsymbol{PN}$ and $\boldsymbol{PT}$
Reverse the backward path, and output the shortest cost path
Output
For each train k output the obtained shortest path

Appendix C:. Proof of Eq. (9)

Property. For any train k, the equality $\sum_a \lambda_a^k \sum_{a' \in U(a)} x_{a'}^k = \sum_{a' \in U(a)} \lambda_{a'}^k \times \sum_a x_a^k$ holds.

Proof. If train $\boldsymbol{k}$ selects arc $\boldsymbol{a}_0 \in \boldsymbol{U}(\boldsymbol{a})$, then $x_{a_0}^k = 1, x_a^k = 0 \boldsymbol{for} \, \boldsymbol{a} \neq \boldsymbol{a}_0$.

$$\begin{aligned}\sum_a \lambda_a^k \sum_{a' \in U(a)} x_{a'}^k &= \sum_a \lambda_a^k \left(\sum_{a_0 \in U(a)} x_{a_0}^k + 0 \right) = \sum_a \lambda_a^k \sum_{a_0 \in U(a)} x_{a_0}^k \\ &= \sum_{a \in U(a_0)} \lambda_a^k x_{a_0}^k = \sum_{a' \in U(a)} \lambda_{a'}^k \sum_a x_a^k\end{aligned} \tag{C.1}$$

Appendix D:. Lagrangian Relaxation algorithm

Algorithm D: Lagrangian Relaxation for Computing Lower Bounds

Input:
$\boldsymbol{G} = (\boldsymbol{N}, \boldsymbol{A})$: Space-time network and arc costs
$\boldsymbol{K}_1$: Set of existing trains (with known schedules)
$\boldsymbol{K}_2$: Set of extra trains (with given time windows, stop patterns, and section running times)
$\boldsymbol{Q}_{LR}$: Maximum iteration count
The initial Lagrangian multipliers $\lambda_a^k(0), \gamma_i^t(0)$
Set iteration counter $\boldsymbol{q} \leftarrow 0$
For $\boldsymbol{q} < \boldsymbol{Q}$:
For each train $\boldsymbol{k} \in \boldsymbol{K}_1 \cup \boldsymbol{K}_2$:
Solve the shortest path problem for train $\boldsymbol{k}$ using the forward dynamic programming algorithm. Assign $x_a^k = 1$ along the obtained shortest path
End for
Calculate the lower bound $\boldsymbol{LB}(\boldsymbol{q})$:
Update the best found lower bound:
$$\boldsymbol{LB}^*(\boldsymbol{q}) = \begin{cases} \boldsymbol{LB}^*(\boldsymbol{q}-1) if LB(\boldsymbol{q}) \leq \boldsymbol{LB}^*(\boldsymbol{q}-1) \\ LB(\boldsymbol{q}) if LB(\boldsymbol{q}) > \boldsymbol{LB}^*(\boldsymbol{q}-1) \end{cases}$$
Update Lagrangian multipliers λ_a^k, γ_i^t using the current solution and the update rules Eq. (21) and Eq. (22)

(*continued on next page*)

(*continued*)

Algorithm D: Lagrangian Relaxation for Computing Lower Bounds
Increment the iteration counter $q \leftarrow q+1$ End for

Data availability

The authors do not have permission to share data.

References

Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society, 48*, 334.

Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). Athena Scientific.

Bešinović, N., Goverde, R., Nicholson, G., & Steele, H. (2021). *Review of railway timetabling developments: Planning phases, goals and future directions. Working paper.* Delft University of Technology.

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.

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.

Brännlund, U., Lindberg, P. O., Nou, A., & Nilsson, J. E. (1998). Railway timetabling using lagrangian relaxation. *Transportation Science, 32*(4), 358–369.

Burdett, R. L., & Kozan, E. (2009). Techniques for inserting additional trains into existing timetables. *Transportation Research Part B: Methodological, 43*(8–9), 821–836. https://doi.org/10.1016/j.trb.2009.02.005

Cacchiani, V., Caprara, A., & Toth, P. (2010). Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological, 44*(2), 215–231. https://doi.org/10.1016/j.trb.2009.07.007

Cacchiani, V., & Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research, 219*(3), 727–737. https://doi.org/10.1016/j.ejor.2011.11.003

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

Caimi, G., Kroon, L., & Liebchen, C. (2017). Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning and Management, 6*(4), 285–312. https://doi.org/10.1016/j.jrtpm.2016.11.002

Caprara, A., Fischetti, M., & Toth, P. (2002). Modeling and solving the train-timetabling problem. *Operations Research, 50*(5), 851–861. https://doi.org/10.1287/opre.50.5.851.362

Caprara, A., Monaci, M., Toth, P., & Guida, P. L. (2006). A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics, 154*(5), 738–753. https://doi.org/10.1016/j.dam.2005.05.026

Chen, L., Hopman, H., & Negenborn, R. R. (2018). Distributed model predictive control for vessel train formations of cooperative multi-vessel systems. *Transportation Research Part C: Emerging Technologies, 92*, 101–118. https://doi.org/10.1016/j.trc.2018.04.013

Corman, F., & Meng, L. (2015). A review of online dynamic models and algorithms for railway traffic management. *IEEE Transactions on Intelligent Transportation Systems, 16*(3), 1274–1284. https://doi.org/10.1109/TITS.2014.2358392

Douglas, J., Jr., & Rachford, H. H., Jr. (1956). On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society, 82*, 421–439. https://doi.org/10.1090/S0002-9947-1956-0084194-4

D'Ariano, A., & Pranzo, M. (2009). An advanced real-time train dispatching system for minimising the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics, 9*(1), 63–84. https://doi.org/10.1007/s11067-008-9088-1

Eckstein, J., & Bertsekas, D. P. (1992). On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming, 55*(1–3), 293–318. https://doi.org/10.1007/BF01581204

Erlandson, W., Häll, C. H., Peterson, A., & Schmidt, C. (2023). Meta-heuristic for inserting a robust train path in a non-cyclic timetable. *Transportation Planning and Technology, 46*(7), 842–863. https://doi.org/10.1080/03081060.2023.2239208

Flier, H., Graffagnino, T., & Nunkesser, M. (2009). Scheduling additional trains on dense corridors. In V. Bonifaci, C. Demetrescu, & A. Marchetti-Spaccamela (Eds.), *Experimental Algorithms (Lecture Notes in Computer Science* (Vol. 5526, pp. 149–160). Springer. https://doi.org/10.1007/978-3-642-02011-7_15.

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.

Gao, Y., Yang, L., & Gao, Z. (2017). Real-time automatic rescheduling strategy for an urban rail line by integrating the information of fault handling. *Transportation Research Part C: Emerging Technologies, 81*, 246–267. https://doi.org/10.1016/j.trc.2017.06.005

Gao, Y., Kroon, L., Yang, L., & Gao, Z. (2018). Three-stage optimization method for the problem of scheduling additional trains on a high-speed rail corridor. *Omega, 80*, 175–191. https://doi.org/10.1016/j.omega.2017.08.018

Gao, R., & Niu, H. (2021). A priority-based ADMM approach for flexible train scheduling problems. *Transportation Research Part C: Emerging Technologies, 123*, Article 102960. https://doi.org/10.1016/j.trc.2020.102960

Ghaemi, N., Cats, O., & Goverde, R. M. P. (2017). Railway disruption management challenges and possible solution directions. *Public Transport, 9*, 343–364. https://doi.org/10.1007/s12469-017-0157-z

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 deDirichlet non linéaires. ESAIM. *Mathematical Modelling and Numerical Analysis, 9*(R2), 41–76.

Hansen, I., & Pachl, J. (2014). *Railway timetabling and operations: Analysis, modelling, optimisation, simulation, performance evaluation.* Hamburg: Eurail Press.

Hassannayebi, E., Zegordi, S. H., & Yaghini, M. (2016). Train timetabling for an urban rail transit line using a Lagrangian relaxation approach. *Applied Mathematical Modelling, 40*(17–18), 9892–9913. https://doi.org/10.1016/j.apm.2016.06.040

Hestenes, M. R. (1969). Multiplier and gradient methods. *Journal of Optimization Theory and Applications, 4*(5), 303–320. https://doi.org/10.1007/BF00927673

Ingolotti, L., Barber, F., Tormos, P., Lova, A., Salido, M. A., & Abril, M. (2004). *An efficient method to schedule new trains on a heavily loaded railway network* (Vol. 3315,, 164–173. https://doi.org/10.1007/978-3-540-30498-2_17

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. https://doi.org/10.1016/j.trb.2017.06.018

Ljunggren, F., Persson, K., Peterson, A., & Schmidt, C. (2021). Railway timetabling: A maximum bottleneck path algorithm for finding an additional train path. *Public Transport, 13*, 597–623. https://doi.org/10.1007/s12469-020-00253-x

Lova, A., Tormos, P., Barber, F., Ingolotti, L., Salido, M. A., & Abril, M. (2007). Intelligent train scheduling on a high-loaded railway network. In C. Lübbecke (Ed.), *Algorithmic Methods for Railway Optimization (Lecture Notes in Computer Science* (Vol. 4359, pp. 219–232). Springer.

Meng, L., & Zhou, X. (2014). Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables. *Transportation Research Part B: Methodological, 67*, 208–234. https://doi.org/10.1016/j.trb.2014.05.005

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. https://doi.org/10.1016/j.trb.2019.02.017

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.

Parbo, J., Nielsen, O. A., & Prato, C. G. (2016). Passenger perspectives in railway timetabling: A literature review. *Transport Reviews, 36*(4), 500–526. https://doi.org/10.1080/01441647.2015.1113574

Powell, M. J. D. (1969). A method for nonlinear constraints in minimization problems. In R. Fletcher (Ed.), *Optimization* (pp. 283–298). London: Academic Press.

Tan, Y. (2013). Bound search algorithm and iterative refinements for scheduling extra train paths. In Transportation Research Board 92nd annual meeting Compendium of Papers. *Transportation Research Board, Paper, 13–3016*, 13 pp.

Tan, Y. (2015). *Techniques for inserting additional train paths into existing cyclic timetables (Doctoral dissertation)*. Braunschweig, Germany: Technische Universität Carolo-Wilhelmina zu Braunschweig.

Tormos, P., Lova, A., Barber, F., Ingolotti, L., Abril, M., & Salido, M. A. (2008). *A genetic algorithm for railway scheduling problems* (Vol. 5011,, 255–276.

Xiu, C., Pan, J., D'Ariano, A., Zhan, S., & Peng, Q. (2024). Passenger service-oriented timetable rescheduling for large-scale disruptions in a railway network: A heuristic-based alternating direction method of multipliers. *Omega, 125*, Article 103040. https://doi.org/10.1016/j.omega.2024.103040

Yao, Y., Zhu, X., Dong, H., Wu, S., Wu, H., Carol Tong, L., & Zhou, X. (2019). ADMM-based problem decomposition scheme for vehicle routing problem with time windows. *Transportation Research Part B: Methodological, 129*, 156–174. https://doi.org/10.1016/j.trb.2019.09.009

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. https://doi.org/10.1016/j.tre.2016.07.015

Zhan, S., Wang, P., Wong, S. C., & Lo, S. M. (2022a). Energy-efficient high-speed train rescheduling during a major disruption. *Transportation Research Part E: Logistics and Transportation Review, 157*, Article 102492. https://doi.org/10.1016/j.tre.2021.102492

Zhan, S., Wong, S. C., Shang, P., & Lo, S. M. (2022b). Train rescheduling in a major disruption on a high-speed railway network with seat reservation. *Transportmetrica A: Transport Science, 18*(4), 532–567. https://doi.org/10.1080/23249935.2021.1877369

Zhan, S., Xie, J., Wong, S. C., Zhu, Y., & Corman, F. (2024). Handling uncertainty in train timetable rescheduling: A review of the literature and future research directions. *Transportation Research Part E: Logistics and Transportation Review, 183*, Article 103429. https://doi.org/10.1016/j.tre.2024.103429

Zhang, Y., Peng, Q., Yao, Y., Zhang, X., & Zhou, X. (2019). Solving cyclic train timetabling problem through model reformulation: Extended time-space network construct and Alternating direction Method of Multipliers methods. *Transportation Research Part B: Methodological, 128*, 344–379. https://doi.org/10.1016/j.trb.2019.08.001

Zhang, Y. (2020). *Optimisation models and algorithms for railway train timetabling problems under complex and typical scenarios*. Chengdu: Southwest Jiaotong University. PhD thesis.

Zheng, Y. J. (2018). Emergency train scheduling on chinese high-speed railways. *Transportation Science, 52*, 1077–1091. https://doi.org/10.1287/trsc.2017.0794