Joint optimization of train platforming and shunting with service scheduling at a railway hub

Abstract

The rapidly developing high-speed railway system in China has posed increasing challenges to train and station scheduling. During rush hours, several switches and tracks are often congested due to the lack of synchronized train platforming and service scheduling. Therefore, we investigate a train platforming and shunting with service scheduling (TPSSS) problem of high-speed railway hub at the tactical level. This joint optimization approach can reduce conflicts, decrease the shunting work, and promote balanced resource utilization at stations and depots. By constructing a two-layer time-space network based on the hub layout, a multi-commodity flow model and a heuristic algorithm that utilizes the alternating direction method of multipliers (ADMM) are developed. Applying an iterative framework that includes inner and outer loops, as well as a pre-assignment strategy, the algorithm can produce a near-optimal (gap < 5%) TPSSS plan within a computation time of 1 hour in cases where there are no cancelled trains. Furthermore, a comparison between conservative and aggressive strategies indicates that the aggressive strategy can achieve an overall better solution.

Keywords

Train Unit Shunting, Service scheduling, Maintenance, Time-space network, ADMM

1 Introduction

In the past decade, China's high-speed railway has developed rapidly, constructing a lot of railway lines and forming a criss-cross high-speed railway network. As multiple lines were gradually put into operation, some hub stations have gone from Sels et al., 2014 belonging to only one line to already connecting multiple lines. However, due to infrastructure limitations, some stations have become bottlenecks that restrict line or network capacity. The majority of delays were caused by conflicts that occur at stations, even resulting in train knock-on delays due to the criss-cross railway network. Therefore, given that the macroscopic train timetable is already provided (Sels et al., 2014), high-quality train platform assignment and in- and out-routes at stations are critical for the implementation of the planned timetable at the station level.

In many cases, the number of *platform tracks* (tracks with platforms for passengers to embark and disembark) at train stations is insufficient for trains to stay overnight, and as a result, shunting yards (also called depots in Haahr et al. (2017) are often located nearby. The process of parking train units (consisting of several carriages) at a shunt yard, together with several related processes, is called shunting (Freling et al., 2005; Kroon et al., 2008). According to van den Broek (2022), a railway hub typically includes a major train station and one or more nearby shunting yards. Fig.1 lists 3 railway hubs, namely, the Eindhoven hub in the Netherlands taken from van den Broek (2022), the Metz-Ville hub in France taken from Kamenga (2020) and the Jinan West hub in Shandong, China. Different from the hub in Europe where the station tracks and shunting yards are connected with many tracks and switches, there is usually a clear distinction between the train station and the shunting yard in China, with only one or two connection lines connecting them, as shown in Fig. 1(c). Shunting yards are typically located about one or a few kilometres away from the train station. In China, the shunting yard is often referred to as a maintenance depot, which also provides trains with overnight parking and low-level maintenance services such as repairing, cleaning, sewage suction, and wheel turning (Xu & Dessouky, 2022). Then, the railway hub in this study based on the Chinese high-speed railway case is defined as the area that includes the station and the maintenance depot.

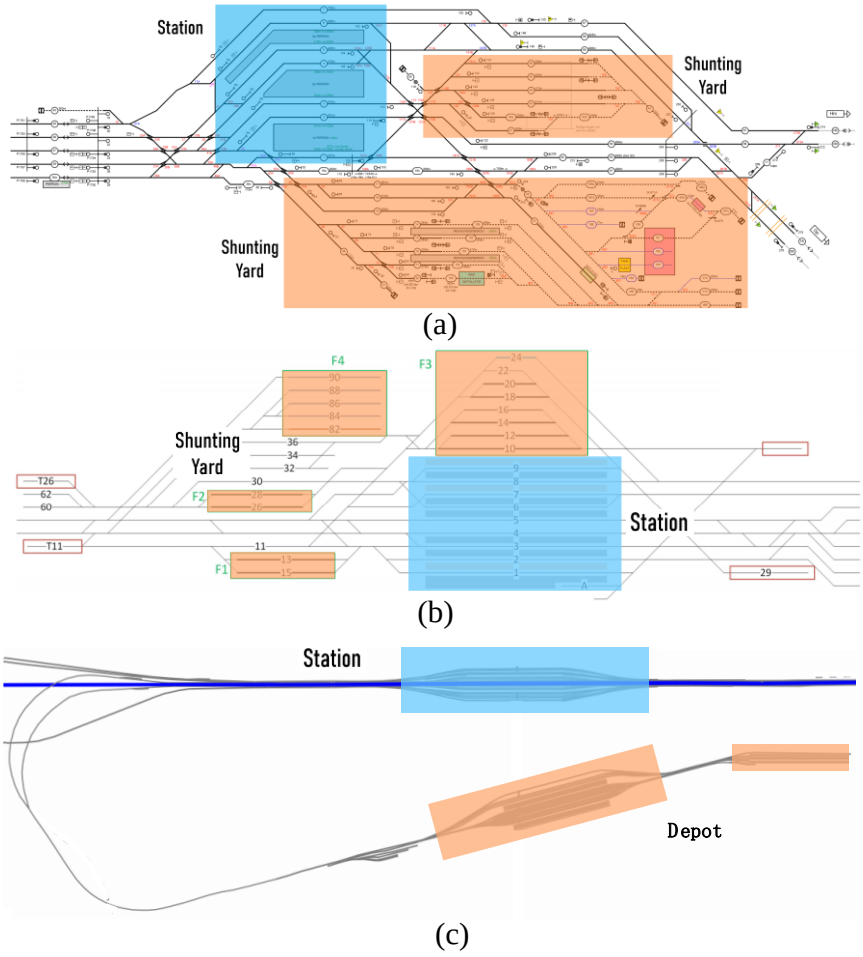


Figure 1: Three railway hubs, (a) Eindhoven hub in the Netherlands, taken from van den Broek (2022), (b) Metz-Ville hub in France taken from Kamenga (2020), (c) Jinan West hub in Shandong, China. The station areas are highlighted in blue, and the shunting yards are highlighted in orange.

A railway hub typically has a vast number of tracks and connects several lines with different directions. In China, a hub may have between 30 to 130 tracks. By the end of 2022, China Railway owned 4,048 standard passenger EMUs (8 railcars), an increase of 37.92% compared to 2017. However, the number of depots, and connection lines in hub have not increased significantly. Therefore, there is a need to design computer-aided programming methods to assist the dispatcher in the planning process. The motivations of this paper are threefold:

(1) It is common that some EMUs fail to complete all the maintenance service tasks within the specified planning horizon, which may lead to delayed departures from the station and necessitate modifications to the original train platforming plan at the station.

(2) Fig.2 illustrates the number of EMUs departing from and entering the depot of the Jinan West Hub throughout the day. The graph takes the shape of a “V”, clearly illustrating the peak of departures in the morning and the peak of arrivals in the evening. During the peak hours for departing from the depot (4:00 to 10:00) and entering the depot (19:00 to 1:00), the number of EMUs accounts for over 60% of the total daily departures and arrivals at the depot. These are typical numbers for many depots in China and all over the world. Thus, the switch areas and the connection lines between the station and the depot can become bottlenecks.

(3) If the shunting yards are not sufficient to park all trains, some trains have to stay longer on the platform tracks, and vice versa. For example, the entry and exit times for non-immediately reverse EMUs with a stop are generally determined by the platform planning dispatcher at the station, but the plan still needs to be approved by the dispatcher at the depot. Especially when capacity is tight during peak hours, the

entry and exit depot times need to be determined through negotiations between the station and the depot dispatchers. If the operation still cannot be scheduled normally after adjustment, feedback needs to be given to the dispatcher to improve the plan by modifying the timetable and the EMU assignment plan. To eliminate the conflicts at the level of shunting operation planning and avoid multiple negotiations, joint planning of the station and the depot assists in balancing the utilization of capacities between the station and the depot.

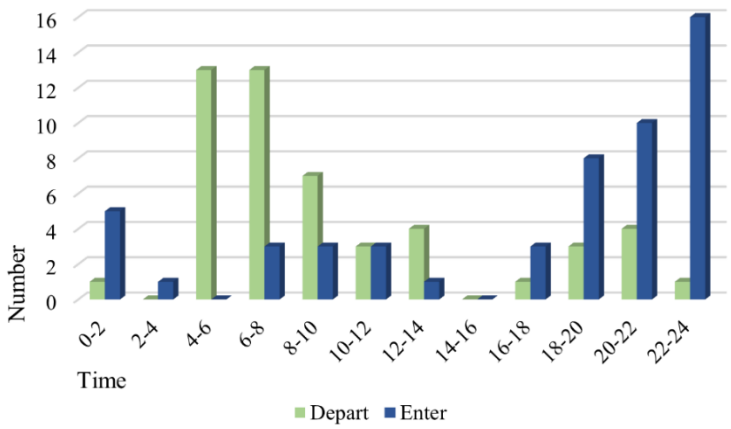


Figure 2: The number of EMUs departing from and entering Jinan Depot throughout the day.

There is also an ongoing pilot project by the Dutch Railways to utilize existing methods to generate the daily operations plan of the whole Eindhoven railway hub, as reported by van den Broek et al. (2022). Taking into account the aforementioned reasons and motivations, this paper aims to jointly optimize train platforming and shunting with service scheduling (TPSSS) at the railway hub on medium-term, e.g., one month, one season, as the China State Railway Group Co., Ltd. (CHINA RAILWAY) adjusts its timetable every quarter each year. The TPSSS plan generated in this paper constitutes a fixed platforming tracks occupation plan for a period of time in the future, until the next timetable adjustment. At this tactical level, the macroscopic timetable is assumed to be fixed, as Sels et al. (2014) did. That is, the arrival and departure times are not allowed to be changed in our problem.

The train unit shunting problem (TUSP) at stations and depots can be divided into three sub-problems: matching, parking, and routing. Train *matching* involves assigning several train units to trains according to the required train composition and unit order. Train *parking* involves arranging suitable tracks for trains. As the title of the ROADEF/EURO Challenge 2014 (Artigues et al., 2018) says, “Trains do not vanish.” Trains must always park or move on the tracks, which is different from the common job shop scheduling problems, as jobs can wait at other places rather than machines in the latter problem. The different capacities and functions of the tracks add a lot of complexity to the problem. Train *routing* mainly involves arranging a path for the train’s movement that has no conflicts or blockages. Additionally, at the depot, EMUs need to undergo a series of maintenance services with no specified order. Determining the order and timing of services is referred to as *service scheduling*. The train unit shunting problem and service scheduling problem are interdependent. Therefore, the entire problem is known as the *train unit shunting problem with service scheduling* (TUSPwSS) (van den Broek, 2022) or generalized train unit shunting problem (Kamenga et al., 2021). It should be noted that the network-level train matching is the outcome of the rolling stock scheduling problem (Wang et al., 2019; Zhong et al., 2019), not considered further in this paper. Considering that train coupling and decoupling rarely occur in China (Xu & Dessouky, 2022; Zhong et al., 2019), the matching subproblem is not considered here. However, the approach in this paper is still suitable for solving the remaining sub-problems, including train parking, train routing and service scheduling. As studied in Freling et al. (2005) and Kamenga et al. (2021) for sequentially solving the TUSP, “the train matching problem is the first problem to be solved in the G-TUSP”. The train matching problem can be instantiated independently, without requiring a solution to other sub-problems.

As the matching is not considered, the input information of the problem in this paper includes train arrival and departure times, train service tasks, and infrastructure information of stations and depots,

including track connectivity, switch layout, and service equipment. The feasible condition of the final plan is meeting the track capacity, avoiding conflicts in the shunting route, and completing all operational tasks including train arrival, departure, repair, cleaning and wheel turning. The output of the problem is the route of each EMU at each time instant to avoid "train vanishing".

The remainder of this paper is structured as follows. Section 2 reviews the related literature and highlights the scientific contributions of this paper. Section 3 provides a detailed description of the problem being considered. In Section 4, the problem is formulated using the time-space network representation. Section 5 presents an ADMM-based heuristic method for solving the problem. Section 6 conducts both an artificial small case study and a real-world case study to validate the effectiveness of our network flow model and solution methodology. Additionally, sensitivity analysis is conducted based on cases of varying scales. Finally, Section 7 concludes the paper and discusses future research directions.

2 State of the art

Many papers have studied the train platforming problem (TPP) at stations, as classified by Caprara et al. (2011), Lusby et al. (2011a). The platforming problem has three levels: (i) The strategic level is for determining or evaluating the capacity of the station; (ii) the tactical level is to assign platform tracks and routes for the fixed macroscopic timetable; (iii) the operational level is to decide the platforms and routes in real time or several hours before the planning horizon. Zwaneveld et al. (2001) study the TPP at the strategic level for timetable evaluation and regarded the TPP as a Node Packing problem. Caprara et al. (2011), Sels et al. (2014) and Zhang et al. (2023) study the tactical TPP. Sels et al. (2014) introduce the dummy platform and dummy route for always generating a feasible solution when using a commercial solver to solve the MILP model. In Caprara et al. (2011), each path links the direction node and the platform track node, and is composed of switch nodes. For a given path, its set of incompatible paths is also given. The modelling in our work also employs this approach to handle conflicts between paths, see Section 4.2. Zhang et al. (2023) integrate the TPP with the locomotive coupling and decoupling operation, and they model the extended TPP based on a sectional interlocking system. They use the time-space-state network to illustrate the whole process, where the "state" dimension is introduced to describe the route-lock sectional-release interlocking mechanism. They mention their extended TPP is similar with the TUSPwSS at depots, as the operation of changing locomotives can be regarded as a service. However, the trains in TUSPwSS stays much longer at the depots, and has much more shunting operations, which makes the TUSPwSS more complex than the extended TPP. (Zhong et al., 2024) study the TPP considering operation scheduling, where each train has an operation task list (called operation chain in their paper) in the station, and the operation sequence is fixed. The operation task list may include arrival, departure, stopping, water supply, sewage suction, entering or leaving the depot and train turnaround. However, the unfixed service sequence and flexible storage operation in our TUSPwSS is more challenging and complex than their extended TPP. For the operational TPP, it is often combined with timetable rescheduling, as seen in previous works Lusby et al. (2013) and Y. X. Zhang et al. (2020). In response to timetable disturbances, rescheduling of platforming plans must be carried out within a short period of time to obtain feasible recovery plans. Since this paper focuses on the tactical-level overall planning of hubs, there is no extreme requirement related to the computation time.

Many researchers have also conducted research on different versions of the TUSP. Jacobsen and Pisinger (2011) use three metaheuristics to jointly solve the train parking and maintenance problem. Haahr et al. (2017) develop three heuristics to solve the parking and matching problem at the railway hub. This is the first paper that performs a joint optimization of stations and shunting yards. van den Broek (2016) considers a complete version of the TUSPwSS, including all sub-problems. A local search heuristic is designed. van den Broek et al. (2022) allow simultaneous routing when performing local search. However, the stability of this local search heuristic algorithm cannot be demonstrated due to the absence of relevant mathematical models. Moreover, the solution quality cannot be theoretically measured as no lower bounds have been obtained. This issue is commonly faced by some heuristic algorithms, which can quickly produce feasible solutions but make it challenging to evaluate their quality, for instance by comparing the feasible solution to a theoretical lower bound. Kamenga (2020) models and solves the TUSPwSS at the hub. He paid more attention to matching. However, the Chinese version of TUSPwSS needs to focus more on service scheduling and parking plan. Concerning European railway systems, Haahr et al. (2017) and Kamenga (2020)

integrated some sub-problems of station platforming and some sub-problems of TUSPwSS. However, they did not consider some important sub-problems, especially the heavy service scheduling work.

Some recent research on train shunting and service scheduling at depots in China are presented in Li et al. (2017), Wang et al. (2021), He et al. (2022) and Xu and Dessouky (2022). Their researches are based on the analysis of a specific layout of the depot, making it difficult to apply to depots with different layouts. Furthermore, their analysis only encompasses first-level maintenance, which involves two service tasks: repair and cleaning. Li et al. (2017) only solve the parking sub-problem at a single yard of a depot. Wang et al. (2021) put forward an ILP model and employed the Gurobi solver to solve it. He et al. (2022) solve the train shunting and service scheduling problem using the PSO metaheuristic. He et al. (2023) incorporate stochastic uncertainties at depots into their methodology. However, both He et al. (2022) and He et al. (2023) suppose all tracks can only accommodate one EMU. Xu and Dessouky (2022) construct a time-space network customized for two services. In addition, there is no literature on the optimization of the operation planning of an entire hub, including both the station tracks and the shunting yard, in China.

Most studies do not consider the train relocation strategy. The train relocation strategy allows the trains to be shunted to another track to avoid conflicts. As tested in van den Broek (2016) and van den Broek et al. (2022), this flexible strategy can help to find feasible plans. This paper allows relocation which means EMUs can be moved to another storage track when parked at the storage track.

In providing an overview, the following papers are summarized within Table 1, where a checkmark (√) denotes the inclusion of a particular aspect.

Table 1: Literature review on TUSP

Reference	Matching	Station platforming		Shunting yard / Depot					Granularity	Objective
		Parking	Routing	Parking	Routing	Servicing	Relocation	Layout		
Caprara et al. (2011)		√	√					General	Microscopic	-
Sels et al. (2014)		√	√					General	Microscopic	-
Zhang et al. (2023)	√	√	√					General	Microscopic	-
Zhong et al. (2024)		√	√			√ **1		General	Microscopic	-
Jacobsen and Pisinger (2011)				√				General	Macroscopic	D,C,M
Haahr et al. (2017)	√	√		√				General	Macroscopic	D
van den Broek (2016)	√			√	√	√	√	General	Macroscopic	D,C,M
van den Broek et al. (2022)	√			√	√	√	√	General	Microscopic	D,C,M
Kamenga (2020)	√		√	√	√	√		General	Microscopic	D
Li et al. (2017)				√				General	Macroscopic	C
Wang et al. (2021)				√	√	√ **2		Specific, typical	Macroscopic	R
Xu and Dessouky (2022)				√	√	√ **2		Specific, typical	Microscopic	TSC
He et al. (2022)				√		√ **2		Specific, typical	Macroscopic	Cmax, G
He et al. (2023)				√		√ **2		Specific, typical	Macroscopic	Cmax, G
This paper		√	√	√	√	√	√	General	Microscopic	TSC

*: Only considering locomotive decoupling and coupling at stations.

**: Each train has a fixed operation sequence in **[1]. Each train has a service task with two services, namely repair and cleaning in **[2].

Objective: D: Minimize the delay time or delayed EMU or cancelled EMU; C: Minimize the number of route conflicts; M: Minimize the number of shunting movements; G: Minimize the occupation time of service tracks; R: Minimize route cost; Cmax: Minimize the completion time of the latest service; C: Maximize the capacity of the depot; TSC: Minimize the cost of the time-space network.

This paper contributes to the existing literature in several aspects. Firstly, to our knowledge, it is the first study to put forward a joint optimization model on the microscopic level to the train platforming and shunting with service scheduling at a railway hub, not limited in depots (Bosi et al., 2024; Li et al., 2017) or limited in stations (Li et al., 2023; Padovano et al., 2024). Second, a two-layer time-space network is designed for the considered problem that allows EMUs to undergo a flexible number of services (including arrival and departure platforming services), change directions at any suitable time and flexible storage parking. The flexible storage parking encompasses two aspects: (i) it allows for storage between two services, which was previously not permitted in Xu and Dessouky (2022); (ii) it allows for the option of relocation of the storage. Third, an ADMM relaxation heuristic to solve the train unit shunting problem with service scheduling is designed. Compared with the framework of the ADMM-based heuristic used in Song and Cheng (2022); Tang et al. (2023); Xue et al. (2023), the heuristic in this paper considers an iterative framework that includes inner and outer loops.

3 Problem description

Within a planning horizon of 12 hours, or more, a set of 8-car or 16-car EMUs pass through the hub with or without stops. EMUs are not allowed to be delayed when arriving or departing from the hub. Some EMUs have maintenance service tasks, which are all in an unfixed sequence. Decisions must be made regarding the platform track where these EMUs will stop, the time for entering and leaving the depot, the schedule of operations for each EMU within the depot, and the parking and routing within the depot, in order to determine a feasible path for each EMU that does not conflict with other EMUs at tracks and switches. For complex problems like the TPSSS, a single objective function may not comprehensively guide optimization computations. Following the approach by Xu and Dessouky (2022), the objective function is defined as the minimization of total operating cost based on a time-space network. The total operating cost includes the parking cost, shunting cost, service cost and penalty cost for cancelling EMUs. Such kind of comprehensive objective function also appears in Xu and Dessouky (2022) and van den Broek et al. (2022). For the model based on the time-space network, the operating cost of one EMU is the total cost of the arcs in its time-space path, and the total operating cost is the sum of operating costs of all EMUs (Xu & Dessouky, 2022). The cost of each kind of arc can be viewed as the weight of different arcs when calculating and differs based on actual service cost or dispatcher's preference. For example, a higher time interval cost to service arcs is set to reduce unnecessary occupation (extra dwelling) of service track, and the fixed penalty cost of a shunting operation is added into the cost function of service arcs and storage arcs to reduce shunting operations. In van den Broek et al. (2022), multiple factors are weighted and summed into the objective function to find the quality of feasible solutions and optimal solutions in the heuristic algorithm, such as the number of delayed EMUs and delay time, the number of route conflicts, the number of over-used tracks, and the number of shunting movements. We also compared the performance when only minimizing total delay time and ignoring all cost using the method in Shi and Li (2019), and found that the comprehensive objective function can improve the quality of the plan, see Appendix D. In He et al. (2022); He et al. (2023), the occupancy time of service tracks is also minimized to increase the utilization rate of service tracks. Moreover, the number of storage tracks is usually much larger than the number of service tracks, so trains are encouraged to stay on storage tracks as much as possible. Shunting preferences of dispatchers and the working hours of maintenance workers also contribute to reducing the occupancy of service tracks. Therefore, in this paper, the cost of staying on services track is set higher than staying on storage tracks, see Section 6.

The assumptions of our study are as follows: (a) Train coupling and de-coupling are not considered in the hub; (b) Adequate service workers and shunting drivers are available within the hub. (c) The station tracks in this paper are assumed to not allow multiple trains to be accommodated on the same platform track

simultaneously. (d) If there exists shunting route(s) linking two adjacent track/position nodes without changing EMU's direction, only one basic shunting route will be considered (Billionnet, 2003; Carey & Carville, 2003; Lusby et al., 2011b; Sels et al., 2014; Q. Zhang et al., 2020; Zhang et al., 2022). The route can be found by using classic graph shortest path search algorithms, such as Dijkstra algorithm (Kavička & Krýže, 2021) or A* algorithm (Riezebos & van Wezel, 2009). A route typically comprises multiple sections, including switches and tracks. Two typical route mechanisms are used in real-world railway yards. The route-locking and route-release interlocking scheme entails locking the entire route once one of its sections is occupied, maintaining the lock until the current occupant vacates it. A more sophisticated and efficient approach, called route-locking and route-release interlocking scheme, involves initially locking a route but permitting a train to claim it once its predecessor clears the shared sections (Lu et al., 2022). In the hub in this paper, the route-locking and route-release interlocking scheme is implemented, which serves as a valid approximation of the sectional-release mechanism (Corman et al., 2009). (e) In order to mitigate the frequent occupancy of service tracks, EMUs are prohibited from accessing service tracks unless they undergo servicing.

According to the sub-problems mentioned in Section 1, the problem is further explained in detail below.

3.1 Train platforming at the station and parking at the depot

Fig.3 illustrates a simplified topology of a railway hub in China. The station is equipped with two main tracks (M1 and M2) for passing, and multiple parallel platform tracks (PT3~PT8) in both directions. The depot is comprised of a cleaning yard, a repair yard, a wheel turning yard and a storage yard. The station and the depot are connected by two connection lines (CL1 and CL2). EMUs park on the storage tracks (S1~S6 in Fig.3) and wait for services and departure. Notably, the cleaning track (C1 in Fig.3) is equipped with a washing machine that can clean the exterior surface of the EMUs as they move slowly. The repair tracks (R1 and R2 in Fig.3) are fundamental resources within the depot for EMU inspection and repair. The wheel turning track (WT1 in Fig.3) features a turning machine specifically designed to inspect the wheel and the damaged wheel surface, to trim the wheel surface into a flat and smooth state and to eliminate wear or unevenness. Once started, a service cannot be interrupted. Additionally, one or more transit tracks (T1) are necessary for train shunting. The red arrows show the whole route for a long EMU arrival platforming at a station and shunting to the depot, parking and shunting within the depot, and shunting back to the station and departing platforming. It can be observed that the EMU arrives at the hub and stays on the platform track PT5. After parking, the EMU moves from the connection line CL1 to reach the depot. After undergoing cleaning and repair services, the EMU is then directed to WT1 for wheel turning using saw-move shunting. Finally, it is parked at S4. After all services and storage, the EMU departs from the depot through the connection line CL2, then parks on platform track PT4. After parking, it leaves the hub.

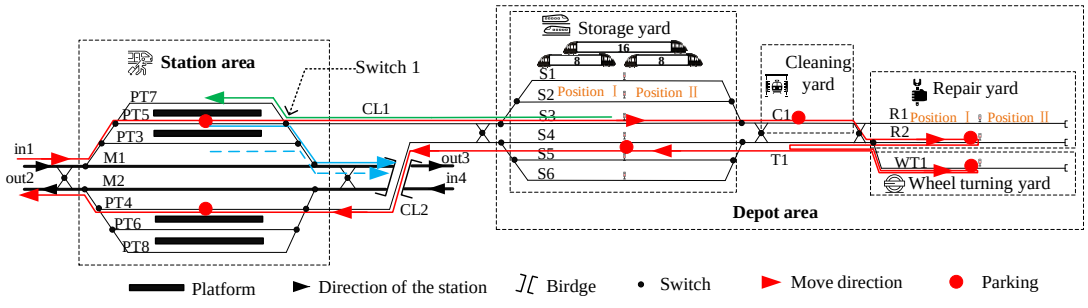


Figure 3: A typical layout of a railway hub in China

The process of train platforming at the station involves both parking and routing decisions. As two linked nodes only have a single route, once the platform is determined, the route is also determined. The conservative strategy, where the platform track and routes of network trains are predetermined, may limit the

available resources for depot trains and result in their inability to arrive, depart or access the depot. For instance, if one network train EMU-1 stops at PT 5 and departs the hub from the out3 node, and at the same time, another depot train EMU-2 needs to leave the depot from CL1 to PT7, the departure route of EMU-1 (blue solid line route in Fig.3) and the shunting route of EMU-2 (green solid line route in Fig.3) conflict with each other in Switch 1 in Fig.3. Under the conservative strategy, the shunting route of EMU-2 should be cancelled, which may cause delay. However, in the progressive strategy, the platforming plan of network trains is not fixed. Thus, if the network EMU-1 is arranged to stop at PT3, then there will be no conflict between the departure route (blue dashed line route in Fig.3) and the shunting route (green solid line route in Fig.3). The stopping time at the platform track of network trains can be considered constant, while the stopping time of depot trains at the platform track after arrival and before departure can be adjusted based on station capacity, depot capacity and the occupancy of connection lines and switches.

Within the depot, the parking decisions include which tracks the EMUs should park on to complete their service tasks, how long of each parking. After completing the current service, such as repair and sewage suction, an extra stay is allowed, which greatly increases flexibility and potential solutions. Moreover, there are only two lengths of EMUs in China, the short EMUs with 8 railcars, and the long EMUs with 16 railcars. Accordingly, the storage track and the repair track at the depot usually have two positions, which can accommodate one long EMU or two short EMUs, as shown in Fig.3. There are five states for each track with double-positions, namely idle, only position 1 occupied, only position 2 occupied, both positions 1 and 2 occupied by different short EMUs, and both positions 1 and 2 occupied by a long EMU. Tracks can be categorized based on the number and direction of entrances and exits. When two short EMUs park at the same track with double-positions, the parking time durations should be carefully designed to avoid blocking when moving.

Different types of trains should be considered for Chinese railway hub: originating trains, terminal trains, through trains without stop, through trains with stop (stopping trains), immediately reverse trains with a stop and non-immediately reverse trains with a stop. The through trains with or without stops and immediately reverse trains that do not enter any shunting yard /depot and only park on platform tracks are called *network trains* in van den Broek (2022). And the remaining trains are called *depot trains* and should enter the shunting yard to park or undergo services. For depot trains, after passengers disembarking the train, the train can continue to stay on the platforming track or it can be shunted to the depot immediately. It is worth noting that determining whether non-immediately reverse trains with stop should enter the EMU depot for parking or remain on the platform tracks of the station depends on the actual situation. If the station is busy, the non-immediately reverse trains with stop can be shunted to the depot, waiting for a long time to depart. The macroscopic timetable and the local route at the station (microscopic timetable) of the network train is usually pre-determined when scheduling the shunting process of depot trains (Kamenga, 2020; van den Broek, 2022). The shunting of depot trains happens at the station area and the shunting yard area and better not affect the network trains. The aim of this paper is to optimize the overall train scheduling process at the station and depot. Two optimization strategies are investigated. The first conservative strategy is to assume that the platforming plan of network trains is given in advance and the plans of all depot trains should not affect the plans of network trains. This strategy is also adopted by Kamenga (2020), Kamenga et al. (2021) and van den Broek (2022). The second strategy, called progressive, is about generating the platforming plan of network trains and the shunting plan of depot trains simultaneously, only based on the given macroscopic timetable.

3.2 Service scheduling

In this paper, the arrival parking and departure parking at the station are also regarded as special services. But it needs to be guaranteed that the arrival parking service is the first (special) service to start, and the departure parking service is the last (special) service to start. This sequence is crucial as the depot services take place subsequent to the EMU's arrival at the station and entry into the depot, and prior to the EMU's departure from the depot and departure parking at the station.

There are two levels of repair service at the depot. The first-level repair is also called routine repair,

which usually takes 60-120 minutes, and the maintenance cycle does not exceed 48 hours or 4000 (5000) kilometers (Lin & Zhao, 2021; Wang et al., 2019). The second-level repair is also called special repair and carried out every 30,000 km or 30 days. The service time ranges from one to several days (Lin & Zhao, 2021; Wang et al., 2019), depending on different running time and mileages. The focus of this paper is short-term service scheduling in the hub during a planning horizon of less than one day, and usually at midnight. However, the second-level repair is usually performed by a special second-level repair maintenance worker team during the daylight hours, so the shunting and service of the second-level repair are usually arranged manually and are not included in this paper. To the best of our knowledge, existing papers on shunting and service scheduling at the depot also do not include second-level repair (He et al., 2022; He et al., 2023; Li et al., 2017; Wang et al., 2021; Xu & Dessouky, 2022). In actual work, if during the planning horizon, for the second-level repair and first-level repair tasks, the long-time second-level maintenance tasks are usually manually arranged first, and then the first-level repair is planned. At this time, the method of this paper can assist the dispatcher in preparing plans.

Fig.4 shows the operation workflow of an EMU at the depot. The EMU should complete all service operations once. Similar to the Open Shop scheduling problem discussed in van den Broek et al. (2022), the service sequence for a single EMU is not predetermined, and should be determined based on the status of other EMUs. This means that the schedule for one EMU may be adjusted based on the availability of resources and the service needs of other EMUs in the depot.

In addition to service scheduling, storage operations may occur between the arrival at the depot and the first service, between two adjacent services, and between the last service and the departure from the depot, as shown in Fig.4. It should be emphasized that these storage operations can be skipped when that leads to better results. That is, the services are necessary and the storage operations are unnecessary. When a storage operation is skipped, the corresponding shunting is also saved. If an EMU requires n services, the number of times the EMU is parked at storage tracks can range from 0 to n+1, and the number of shunting times can range from n+1 to 2n+2, depending on whether the equipment is idle or not. As a result, there are a total of $n! \times 2^{n+1}$ possible operation processes. The traditional modelling approach typically involves setting 0-1 variables to indicate the order between adjacent operations. However, the exponential growth of possible operation processes poses a challenge to this approach. In addition, it is important to minimize unnecessary storage operations to reduce the additional shunting operations.



Figure 4: The operation workflow of an EMU at the depot

3.3 Train routing

To ensure that trains are serviced correctly and to avoid track/position or switch conflicts, shunting movements must be carefully planned. Switch conflicts can occur when two consecutive routes share the same switch segment, and the safety headway time for switches is not met. The interaction between the route at the station and the route at the depot is influenced by the shunting route via the connection line. The conflict point between the routes of EMU entering and exiting the main line and the depot typically occurs at the junction of the connection line with the station's platforming tracks or the main lines. Such route conflicts necessitate manual adjustments by dispatchers and can lead to train arrival or departure delay, particularly during peak hours as mentioned in Section 1. For instance, the shunting route from the depot to the platform track at the station, as indicated by the green solid line arrows in Fig.3, shares a common switch with the blue solid line route. Therefore, to avoid conflicts, it is imperative to resolve the route conflicts either temporally or spatially, by adjusting the start time of the green solid line route or converting the blue solid line route to the blue dashed line route. These issues can be addressed through microscopic (Kamenga, 2020; Xu & Dessouky, 2022) or macroscopic modelling approaches (Billionnet, 2003; Q. Zhang et al., 2020; Zhang et al., 2022). To address tactical-level problem solving, route conflicts on the microscopic level in

switch areas are addressed by employing a pre-processing step before the modelling phase, inspired by Billionnet (2003), Caprara et al. (2011) and Zhang et al. (2022).

This reversal shunting of direction is referred to as a saw-move (van den Broek, 2016), such as from track R2 to track WT1 as illustrated in Fig.3. The EMU must change its direction at transit track T1. Saw-moves are time consuming and require extra walk by the driver, and should only be executed when necessary. Therefore, additional penalty costs and walking time should be taken into account for saw-moves.

Crossing the main line often has adverse effects, e.g., adding potential danger to train flow on the main line. On the one side, some crossings of the main line are unavoidable, e.g., for the immediately reverse trains with a stop at the station. For example, in Fig.3, the EMU travels from direction in1, and stops at platform track PT3. When it is time for the EMU to depart, the EMU crosses the main line and then departs in the direction of out2. On the other side, some crossings of the main line can be avoided thanks to jointly planning, see the example in Section 6.3.2. The cost of crossing the main line is set to a higher value to reduce such crossing.

4 Time-space network formulation

In recent years, the optimization of the railway scheduling problems (Caprara et al., 2002; Wang et al., 2022) and the vehicle routing problem (Mahmoudi & Zhou, 2016; Yao et al., 2019) has seen widespread utilization of the time-space network or time-space-state network formulation. This section describes the construction of a time-space network by discretizing time for each physical node, resulting in a time-space network $G = (V, A)$ for each EMU. The constraints of the movement process can be embedded in the network, such as the arrival time, the departure time, the standard service time duration, the connection relationship between tracks and the state (shunting or dwelling) of EMU. This can help simplify the model of this complex problem. Section 4.1 and Section 4.2 provide detailed explanations of how the physical network of the infrastructure resource and the time-space network is constructed, while Section 4.3 demonstrates the incompatible arc set for nodes and switches. Finally, Section 4.4 presents the overall model.

4.1 Physical network

To facilitate the integrated modelling of TPSSS problem, the physical railway network representation as described in Zhang et al. (2022) is adopted. Fig.5 illustrates the process of nodalization of tracks / positions for the example hub depicted in Fig.3. The storage tracks and repair tracks typically feature two positions for short EMUs, which are referred to as *position nodes* with suffixes with “-1” and “-2”, and are depicted as black squares in Fig.5. For instance, black square “t1-2” indicates Position 2 of Track 1. Tracks equipped with a double-position configuration can accommodate either two short EMUs simultaneously or one long EMU. *Integrated track nodes* (blue squares in Fig.5) with suffix with “-0” are added to these tracks to represent the entire track with two positions where long EMUs will park. For tracks without two positions, they are considered *track nodes* (red squares in Fig.5). Additionally, in and out nodes are established for train arrival and departure (yellow squares in Fig.5). Ultimately, all squares marked in Fig.5 form the physical nodes of the EMU depot.

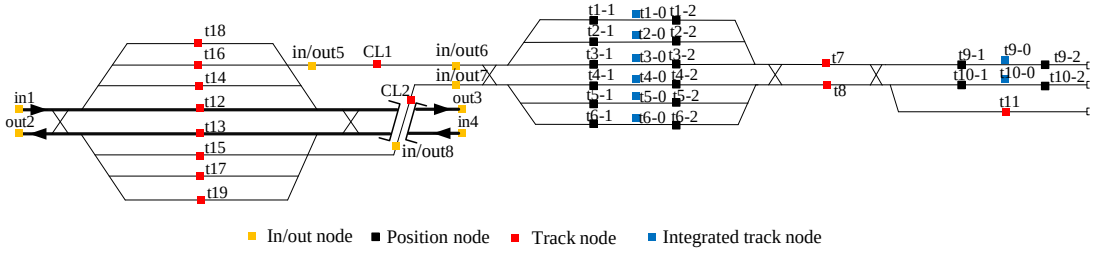
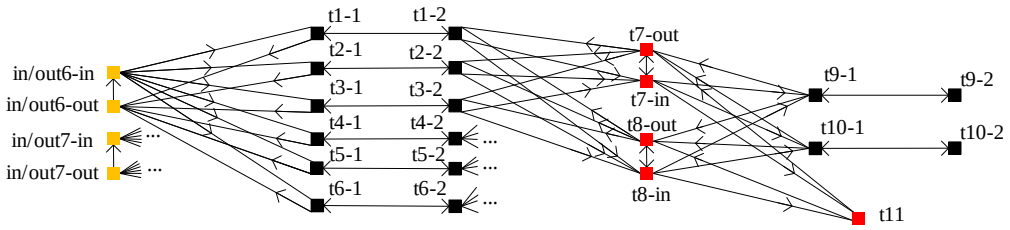


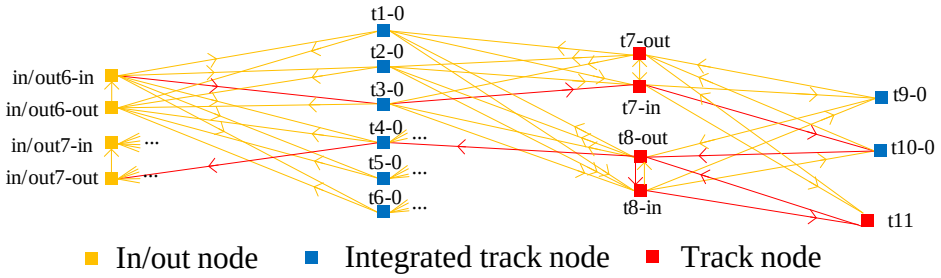
Figure 5: The process of nodalization of tracks / positions

Track occupancy can be classified into two types: *static parking*, where EMUs occupy tracks without being in motion, such as during repair or storage; and *non-static parking*, which involves slow movement of EMUs, as seen in processes like shunting through the track (i.e., node t8) or driving through the fixed cleaning machine on the cleaning track (i.e., node t7). In instances where static parking is prohibited at node i, yet saw-moves are allowed (e.g., at transit track nodes, cleaning nodes, and in/out nodes), node i is substituted with two dummy saw-move nodes: i-in and i-out. This arrangement facilitates the distinction of movement direction. If static parking is permitted at node i, EMUs can also change their direction after parking. In this situation, no saw-move nodes are needed, because the parking duration is sufficient to perform the saw-move.

For the long EMUs, position nodes are removed from their available nodes list. For the short EMUs, integrated track nodes are removed from their available nodes list. Based on connectivity and movement direction restrictions between track nodes, by connecting available nodes with directional or bidirectional arcs, the physical network for short and long EMUs at the example depot can be obtained in Fig.6. Each arrow in the physical network in Fig.6 describes the feasible movement directions. It can be found that the arc between the saw-move node (i.e., i-in and i-out) and the other node is unidirectional, which limits the movement direction. Considering that the arrival/departure time, arrival/departure direction, EMU length, service tasks, etc. of different EMUs are different, defining available nodes and building different physical networks for each EMU based on the EMU length and the arrival/departure direction at the physical network level can reduce the size of the time-space network available to each EMU. For instance, the example route at the depot of a long EMU depicted in Fig.3 is in/out6-in →t3-0 →t7-in →t10-0 →t8-out →t8-in →t11 →t8-out →t4-0 →in/out7-out, as shown in red arrows in Fig.6, where the saw-move route is t10-0 →t8-out →t8-in →t11.



(a): physical network for short EMUs



(b):physical network for long EMUs

Figure 6: The physical network for short and long EMUs at the example depot

4.2 Time-space network

A summary of notations used is given in Table 2. Obviously, each node i in the network allows shunting of EMUs. Some of the nodes, also allow dwelling. Those nodes are transformed into two separate nodes, i^1 and i^2, each representing a different state (*shunting* and *dwelling*), representing the "space"

dimension of the time-space network. In the time-space network, the planning horizon is discretized into a set of time instants. In this paper, time is discretized into one-minute intervals. The time-space network is illustrated in Fig. 7, Fig. 8 and Fig. 9, with a vertex set V and an arc set A. Once costs are assigned to the space-time arcs, the TPSSS problem is transformed into a typical multi-commodity network flow problem. Multi-commodity refers to multiple items that need to be allocated or transported efficiently across a network. The movements of an EMU within the depot can be conceptualized as the flow path of a commodity within a network. The scheduling plan can then be formulated by computing the least-cost flow.

The time space network in this paper is a two-layer directed acyclic time-space network. The two layers represent the shunting state and the dwelling state, and a service results in going from the shunting state (layer) to the dwelling state (layer). This is further illustrated in Fig. 7, Fig. 8 and Fig. 9. Since the time dimension is always forward, the network is acyclic, and this property allows the use of some classical minimum-cost flow search algorithms.

Table 2: Definition of notations

Set	
E	Set of EMUs, e as the index
E^{DS}	Set of EMUs that need to undergo depot services
E^{DP}	Set of EMUs that must enter the depot, $E^{\mathrm{DS}} \subseteq E^{\mathrm{DP}}$
E^{NIR}	Set of non-immediately reverse EMUs. $E^{\mathrm{NIR}} \cap E^{\mathrm{DP}} = \emptyset$
N^1	Set of shunting nodes at the shunting layer, $i^1 \in N^1$
N^2	Set of dwelling nodes at the dwelling layer, $i^2 \in N^2$
N	Set of nodes. $N = N^{\mathrm{in}} \cup N^{\mathrm{out}} \cup N^{\mathrm{pos}} N^{\mathrm{int-tra}} \cup N^{\mathrm{tra}}$, where N^{in} is the set of entrance nodes, N^{out} is the set of exit nodes, N^{pos} is the set of position nodes, $N^{\mathrm{int-tra}}$ is the set of integrated track nodes, N^{tra} is the set of track nodes. Also, $N = N^1 \cup N^2$.
N^{sto}	Set of nodes for storage
N^e	Set of nodes of EMU e. For short EMU e, $N^e = N \backslash N^{\mathrm{int-tra}}$, and for long EMU e, $N^e = N \backslash N^{\mathrm{pos}}$
T	Set of time instant, t as the index
V	Set of vertices, $V = \{\bar{o}, \bar{d}\} \cup \{(i^w, t) \mid i^w \in N, t = 0,1,\dots,T, w = 1,2\}$, where $\bar{o}$ denotes the dummy origin vertex, and $\bar{d}$ denotes the dummy sink vertex, and (i^w, t) represents a node i in state w (shunting or dwelling) at time t.
B	Set of critical switches of the depot, b as the index
P	Set of service type, p as the index
P^e	Set of service type of EMU e, p as the index
A	Set of time-space arcs, $(n,t) \to (m,\tau)$ or $u \to v$ or a as the index, The space-time arc $(n,t) \to (m,\tau)$ denotes the link between the vertex (n,t) and the vertex (m,τ) in the time-space network.
A^e	Set of arcs in the time-space network of EMU e
A^e_{plat}	Set of service arcs of platforming in the time-space network of EMU e
A^1_{nt}	Incompatible arc set for vertex (n,t), $n \in N^2 \backslash N^{\mathrm{int-tra}}$
A^2_{nt}	Incompatible arc set for vertex (n,t), $n \in N^1 \backslash N^{\mathrm{int-tra}}$

A_{bt}	Incompatible arc set for critical switch b at time instant t, $b \in B$
Ψ_n	Incompatible node set for node n

Parameter	
(n_e^a, a_e)	The starting vertex of EMU e, where n_e^a is the arrival node and a_e is the arrival time of EMU e
(n_e^d, d_e)	The ending vertex of EMU e, where n_e^d is the departure node and d_e is the departure time of EMU e
pre_d_e	For EMU e entering the station from the depot, the maximum time duration that is allowed for early entry into the station.
t_{ij}	The time traversing through the physical arc $i \to j$
t_n^{h}	The safety headway time between two EMUs passing through the same node $n \in N$
t_b^{h}	The safety headway time between two EMUs passing through the same switch $b \in B$
c_{ea}	Cost of arc a for EMU e
t_{ep}^{ser}	Standard time of service type p for EMU e, Specifically, the standard time of departure platforming service for EMU e is denoted as $t_{e,\mathrm{dplat}}^{\mathrm{ser}}$.
p_n	Service type of node n

Decision Variable	
$x_{u,v}^e$	=1 if arc $u \to v$ is traversed by EMU e, =0 otherwise.

In the two-layer time-space network of EMU e, there are the following types of arcs:

(1) starting arc

The network starts from a dummy origin vertex $\bar{o}$. There is one starting arc $a = \bar{o} \to (n_e^a, a_e)$ for an EMU to enter the network, with the cost $c_{ea} = 0$.

(2) ending arc

The network end to a dummy sink vertex $\bar{d}$. There is one ending arc $a = (n_e^d, d_e) \to \bar{d}$ for an EMU to depart from the network, $c_{ea} = 0$.

(3) shunting arc

The set of shunting arcs is denoted as A^{shu}. For any $i^1, j^1 \in N^1 \cap N^e$, if there exist physical directional links between node i^1 and node j^1, shunting arcs can be defined as $a = (i^1, t) \to (j^1, t')$, where $t' = t + t_{ij}$, $t \geq a_e$, $t' \leq d_e$, $c_{ea} = c_{ea}^{\mathrm{shu}} t_{ij}$. The cost parameter c_{ea}^{shu} is the cost per time interval of shunting arc a for EMU e. Traversing the shunting arc means to pass through a physical link, i.e., t10-0[1] →t8-out[1] in Fig.6, and t_{ij} is equal to the time duration of the movement. Specifically, if i^1 and j^1 are the saw-move nodes of the same physical node, the arc $(i^1, t) \to (j^1, t')$ is called saw-move shunting arc, and t_{ij} is equal to the walking time from one side to the other side and preparing time of drivers, and $c_{ea} = c^{\mathrm{fix}}$. The cost parameter c^{fix} is the fixed cost per shunting operation. Due to the train length, the t_{ij} of long and short EMUs is different.

(4) service arc

The set of service arcs is denoted as A^{ser}. For any $i^1 \in N^1 \cap N^e \backslash N^{\mathrm{sto}}$, $i^2 \in N^2 \cap N^e \backslash N^{\mathrm{sto}}$, there are service arcs $a = (i^1, t) \to (i^2, t')$, $t' = t + t_{ep_{i^1}}^{\mathrm{ser}}$, $t \geq a_e$, $t' \leq d_e$, $c_{ea} = c_{ep_{i^1}}^{\mathrm{ser}} * t_{ep_{i^1}}^{\mathrm{ser}} + c^{\mathrm{fix}}$. The cost parameter $c_{ep_{i^1}}^{\mathrm{ser}}$ is the cost per time interval of service arcs of service type p_{i^1} for EMU e. The arrival platforming and departure platforming with the standard duration can be regarded as special services. The parameter c^{fix} represents the fixed penalty cost of a shunting operation, which requires a dedicated driver to

operate in practice.

(5) storage arc

The set of storage arcs is denoted as A^{sto}. For any $i^1 \in N^1 \cap N^e \cap N^{\text{sto}}$, $i^2 \in N^2 \cap N^e \cap N^{\text{sto}}$, there are storage arcs $a = (i^1, t) \to (i^2, t)$, $a_e < t < d_e$, $c_{ea} = c^{\text{fix}}$.

(6) dwelling arc

The set of dwelling arcs is denoted as A^{dwe}. For any $i^2 \in N^2 \cap N^e$ and i^2 is allowed for dwelling, there are dwelling arcs $a = (i^2, t) \to (i^2, t+1)$, $a_e < t < d_e - 1$. If $i^2 \notin N^{\text{sto}}$ is allowed for extra dwelling after service, $c_{ea} = c^{\text{dwe}}_{ep_{i^2}}$; if $i^2 \in N^{\text{sto}}$, $c_{ea} = c^{\text{dwesto}}_e$, where $c^{\text{dwe}}_{ep_{i^2}}$ represents the cost per time interval of dwelling arcs of service type p_{i^2} for EMU e, and c^{dwesto}_e is the cost per time interval of dwelling arcs of storage. When the EMU chooses to park at the track for one more time interval after service/storage, it will pass through the dwelling arc.

(7) state transfer arc

For any $i^2 \in N^2 \cap N^e$, $i^1 \in N^1 \cap N^e$, there are state transfer arcs $a = (i^2, t) \to (i^1, t)$, $a_e < t < d_e$, $c_{ea} = 0$. When a state transfer arc is traversed, it indicates the completion of parking and the start of shunting.

(8) dummy arc

For any $i^1 \in N^{\text{in}} \cap N^e$, $j^1 \in N^{\text{out}} \cap N^e$, there are dummy arcs $a = (i^1, a_e) \to (j^1, d_e)$. If an EMU traverses the dummy arc, it indicates that the operations of this EMU cannot be executed due to the limited capacity of the hub. Let $c_{ea} = c^{\text{dwesto}}_e * (d_e - a_e) + \sum_{p \in P^e} (c^{\text{ser}}_{ep} * t^{\text{ser}}_{ep} + c^{\text{fix}})$.

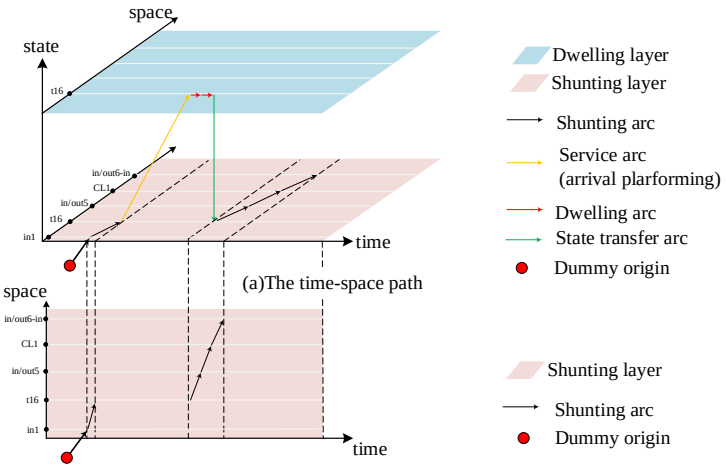
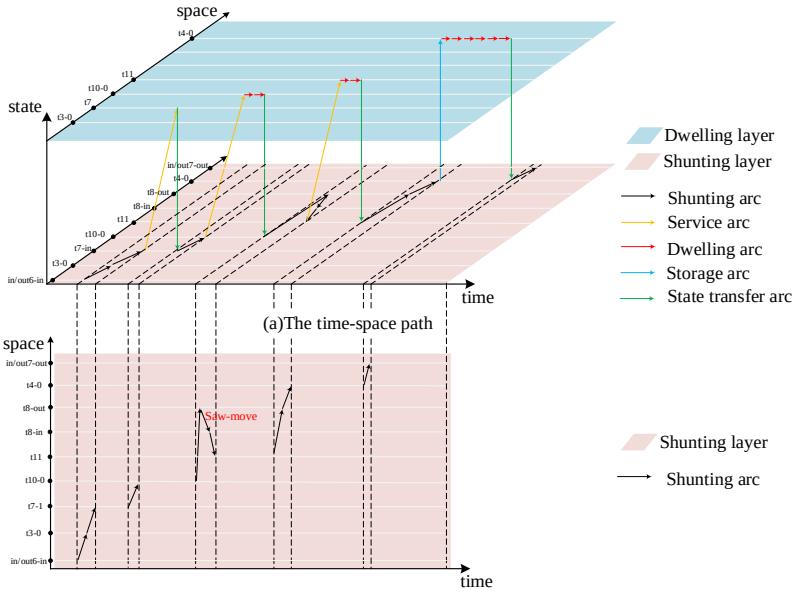
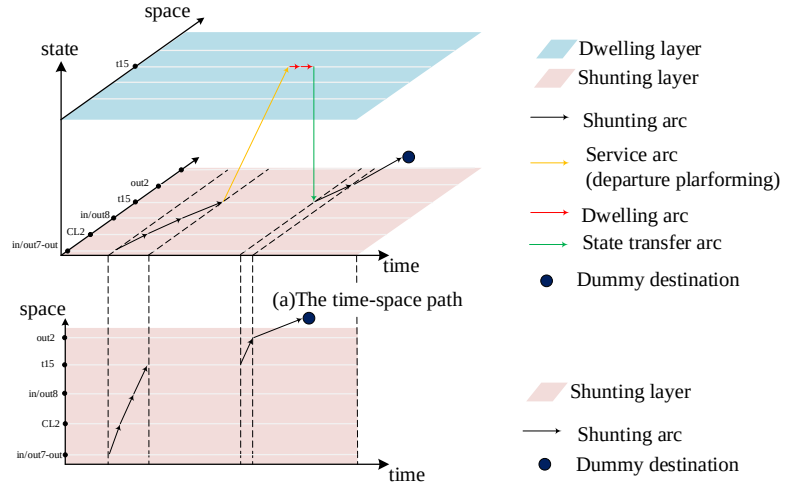


Figure 7: The time-space path of the arrival platforming and shunting of a long EMU



(b)The projection of time-space path on the shunting layer.

Figure 8: The time-space path of a long EMU at the depot



(b)The projection of time-space path on the shunting layer.

Figure 9: The time-space path of the shunting and departure platforming of a long EMU

Fig. 7(a), Fig. 8(a) and Fig. 9(a) respectively illustrate the time-space process for a long EMU arrival platforming at a station and shunting to the depot, parking and shunting within the depot, and shunting back to the station and departing platforming, as depicted in the example route in Fig.3. Each layer refers to a state, inspired by the time-space-state network in Mahmoudi and Zhou (2016). Fig. 7(b), Fig. 8(b) and Fig. 9(b) depict the projection of time-space paths on the shunting layer to show the shunting location more intuitively and clearly. The time span of the yellow service arc is equal to the standard time of service. When a service arc is traversed, it indicates the completion of a service operation. Traversing the service/storage arc means to enter a service/storage track. After the service is completed, the red dwelling arc indicates that it continues to stay on the current track, e.g., extra parking at the platforming track in Fig. 7 and at the repair track in Fig. 8.

Under the assumption that EMUs cannot access service tracks without undergoing services, the utilization of a single-layer space-time network may result in EMUs occupying service tracks for additional

short-term shunting. Consequently, there is no assurance regarding the sequence of shunting EMUs first (shunting arcs), followed by service or storage (service/storage arcs), and ultimately allowing extra dwelling (dwelling arcs). To mitigate this issue, the dwelling arc is assigned subsequent to the service/storage arc, ensuring that service/storage consistently precedes dwelling. This sequential arrangement establishes a coherent link between the two-layer networks, as depicted in Fig. 7(a), Fig. 8(a) and Fig. 9(a). Furthermore, the number of shunting can be equated with the frequency of transitions from the shunting layer to the dwelling layer. The inclusion of the fixed cost associated with each shunting operation into the cost of the service arc/storage arc addresses this challenge effectively. Hence, the implementation of a two-layer network framework serves to overcome this obstacle.

In comparison to the two-layer network proposed by Xu and Dessouky (2022), our network has several improvements: (i) Our approach expands the time-space network to the station, thus enabling support for the planning of operations and shunting of the entire hub; (ii) Our network allows EMUs to have multiple service tasks (including arrival and departure platforming services) with different sequences, instead of their tailored network and model for just two services in Xu and Dessouky (2022). Indeed, both in China and in other countries, train depots offer more than just two types of services; (iii) Our network allows trains to change direction multiple times by incorporating saw-move nodes. This enables our model to be applicable to more complex depots and service tasks. In contrast to Xu and Dessouky (2022), the fixed penalty cost of shunting is also taken into account to reduce unnecessary shunting; (iiii) the storage parking in our network can exist between any two adjacent services, but it can also be avoided. Relocation is also allowed, which helps to get a feasible plan (van den Broek et al., 2022). This will be discussed further in Section 6.1.

4.3 Incompatible arc set for nodes and switches

Two time-space arcs are considered incompatible if both corresponding physical arcs share the same micro-segment, also referred to as track circuits, and the headway time requirement is not met. A track circuit is the smallest unit of railway infrastructure used to detect the presence of trains by electrical current or electromagnetic induction. Conflicts between physical arcs are judged by determining whether they share the same track circuit at the microscopic level. To restrict resource usage, incompatible time-space arc sets A_{nt}^1, A_{nt}^2 and A_{bt} are established for nodes and critical switches, using the method in the previous research (Shi et al., 2024).

4.4 Model formulation

To simplify notation, let $T_n' = T - t_n^{\mathrm{h}} + 1$, $T_b' = T - t_b^{\mathrm{h}} + 1$. To solve the TPSSS problem, a 0-1 integer programming model **M** is constructed.

$$\mathrm{Min} \sum_{e\in E} \sum_{u\to v\in A^e} c_{u\to v}^e \, x_{u\to v}^e \tag{1}$$

s.t

$$\sum_{v:\bar{o}\to v\in A} x_{\bar{o},v}^e = 1, \forall e \in E \tag{2}$$

$$\sum_{u:u\to \bar{d}\in A} x_{u,\bar{d}}^e = 1, \forall e \in E \tag{3}$$

$$\sum_{u:u\to v\in A} x_{u,v}^e - \sum_{w:v\to w\in A} x_{v,w}^e = 0\,, \forall v \in V\backslash\{\bar{o},\bar{d}\}, \forall e \in E \tag{4}$$

$$\sum_{u,v\in A^e\cap A_{\mathrm{plat}}^e : t_u < a_e+5} x_{u,v}^e = 1, \forall e \in E \tag{5}$$

$$\sum_{u,v\in A^e\cap A_{\mathrm{plat}}^e : t_u > d_e - pre_d_e - t_{e,\mathrm{dplat}}^{\mathrm{ser}}} x_{u,v}^e = 1, \forall e \in E^{\mathrm{DP}} \tag{6}$$

$$\sum_{u,v\in A^e\cap A_{\mathrm{plat}}^e : t_u > d_e - pre_d_e - t_{e,\mathrm{dplat}}^{\mathrm{ser}}} x_{u,v}^e \le 1, \forall e \in E^{\mathrm{NIR}} \tag{7}$$

$$\sum_{u\to v\in A^e\cap A_p^{\mathrm{ope}}} x_{u,v}^e = 1, \forall e \in E^{\mathrm{DS}}, \forall p \in P \tag{8}$$

$$\sum_{e\in E} \sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e \le 1, \forall n \in N^2\backslash N^{\mathrm{int-tra}}, t = 0,1,\dots,T_n' \tag{9}$$

$$\sum_{e\in E}\sum_{u\to v\in A^e\cap A^2_{nt}} x^e_{u,v} \le 1, \forall n \in N^1\backslash N^{\text{int-tra}}, t = 0,1,\dots,T'_n \tag{10}$$

$$\sum_{e\in E}\sum_{u\to v\in A^e\cap A_{bt}} x^e_{u,v} \le 1, \forall b \in B, t = 0,1,\dots,T'_b \tag{11}$$

$$x^e_{u,v} \in \{0,1\}\ \forall u \to v \in A, e \in E \tag{12}$$

The objective function (1) is to minimize the total operating cost, which is mentioned at the start of Section 3. It includes the cost of different types of arcs. Constraints (2) (3) (4) are the flow balance constraints. Constraints (5) ~ (8) are service times constraints for each type of service, where t_u means the time dimension of vertex u. Constraints (5) indicate that each EMU should receive an arrival platforming service immediately (immediately after arriving at the platform track, but the actual time is not fixed, as the route duration is not the same. Here, an appropriate small route time spent can be acceptable, for instance, less than 5 time units) after arrival. Constraints (6) guarantee that each EMU from the depot to the station should receive a departure platforming service. EMUs are allowed to enter the depot later and leave the depot earlier, that is, stay more time at the platform track. EMUs are allowed to enter the depot later and leave the depot earlier, that is, stay more time at the platform track. Constraints (7) allow each non-immediately reverse EMU to choose between parking at the depot or continuously parking on the platform track until the departure time. For the latter option, EMU does not need the second platforming service. Constraints (8) indicate that the service task of one EMU needs to be performed exactly once. "Clique" constraints (9) ~ (11) are incompatibility constraints considering the track capacity and headway requirements of tracks and switches. According to the formulas of A^1_{nt}, A^2_{nt}, see Shi et al. (2024), when $m \in N^{\text{pos}}, n \in N^{\text{int-tra}}$, and m, n belong to the same track, i.e., $m = t3 - 1^1$, $n = t3 - 0^1$, then $A^1_{nt} \subset A^1_{mt}$. Therefore, the set $N^{\text{int-tra}}$ is excluded in constraints (9) ~ (11). By setting different sets of arcs for EMUs, long EMUs and short EMUs have different available infrastructure resources. Together with the resource usage limit constraints (9) ~ (11), the parking conflicts of tracks/positions and routing conflicts are eliminated. Constraints (12) define the binary decision variables.

5 Solution Method

Unfortunately, due to the two-layer replication of physical nodes and the discretization of time, the time-space network contains a large number of arcs. Through testing on the smallest instance on the small virtual hub in Section 6.1, it was found that neither the Cplex nor Gurobi solver could obtain a feasible solution within 2 hours. Therefore, it is a common practice to leverage the properties of time-space networks for solving such problems. In this section, a heuristic method based on the Alternating Direction Method of Multipliers (ADMM) is presented. The ADMM method has been used successfully in multi-commodity network flow problems (Yao et al., 2019; Zhang et al., 2019). We will first illustrate how to decompose our model into subproblems of a single train by Lagrangian relaxation (LR) and ADMM and then present the time-dependent label setting algorithm and the upper bound heuristic.

5.1 Problem decomposition based on LR

LR is a commonly used dual decomposition approach that can simplify multi-commodity network flow models (Tang et al., 2023; Wang et al., 2022). This approach relaxes the coupling constraints (hard constraints) and adds them to the objective function as penalty terms. After decomposition, the original model can be divided into several single-commodity network flow subproblems, which can be easily solved.

First, the original model is modified by relaxing the "clique" constraints (9) ~ (11) and introducing non-negative Lagrangian multipliers λ^1_{nt}, λ^2_{nt}, λ^3_{bt}. The resulting LR model $\mathbf{M}_{\text{LR}}$ can be expressed as follows:

$$\begin{aligned}\min Z_{\text{LR}}(x,\lambda) = \sum_{e\in E}\sum_{u\to v\in A^e} c^e_{u,v}\, x^e_{u,v} + \sum_{n\in N^2\backslash N^{\text{int-tra}}}\sum_{t\in T'_n}\lambda^1_{nt}\cdot R^1_{nt} + \\ \sum_{n\in N^1\backslash N^{\text{int-tra}}}\sum_{t\in T'_n}\lambda^2_{nt}\cdot R^2_{nt} + \sum_{b\in B}\sum_{t\in T'_b}\lambda^3_{bt}\cdot R^3_{bt},\end{aligned} \tag{13}$$

where

$$R_{nt}^1 = \sum_{e\in E}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e - 1, \forall n \in N^2\backslash N^{\text{int-tra}}, t \in T_n';$$

$$R_{nt}^2 = \sum_{e\in E}\sum_{u\to v\in A^e\cap A_{nt}^2} x_{u,v}^e - 1, \forall n \in N^1\backslash N^{\text{int-tra}}, t \in T_n';$$

$$R_{bt}^3 = \sum_{e\in E}\sum_{u\to v\in A^e\cap A_{bt}} x_{u,v}^e - 1, \forall b \in B, t \in T_b',$$

subject to constraints (2) ~ (8), and (12).

To express the objective function in a more convenient form, we can rewrite objective function (13) as equation (14), where λ_n refers to the n-th relaxed constraint.

$$Z_{\text{LR}} = \sum_{e\in E}[Z_{\text{LR}}^e] - \sum_n[\lambda_n]. \tag{14}$$

where

$$\begin{aligned} Z_{\text{LR}}^e = \sum_{u\to v\in A^e} c_{u,v}^e\, x_{u,v}^e + \sum_{n\in N^2\backslash N^{\text{int-tra}}}\sum_{t\in T_n'} \lambda_{nt}^1 \cdot \sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e + \\ \sum_{n\in N^1\backslash N^{\text{int-tra}}}\sum_{t\in T_n'} \lambda_{nt}^2 \cdot \sum_{u\to v\in A^e\cap A_{nt}^2} x_{u,v}^e + \sum_{b\in B}\sum_{t\in T_b'} \lambda_{bt}^3 \cdot \sum_{u\to v\in A^e\cap A_{bt}} x_{u,v}^e. \end{aligned} \tag{15}$$

By decomposing the original problem, independent sub-problems specific to each train can be derived. Within the LR framework, these sub-problems can be computed in parallel, allowing for an efficient solution approach. According to the incompatible arc set, after rewriting obj. (15), the model $\mathbf{M}_{\text{LR}}^e$ of the sub-problem for one EMU e becomes:

$$Z_{\text{LR}}^e = \sum_{u\to v\in A^e} \tilde{c}_{u,v}^e x_{u,v}^e. \tag{16}$$

where

$$\tilde{c}_{u,v}^e = \begin{cases} c_{u,v}^e + \sum_{n,t:u\to v\in A^e\cap A_{nt}^1} \lambda_{nt}^1 + \sum_{n,t:u\to v\in A^e\cap A_{nt}^2} \lambda_{nt}^2 + \sum_{b,t:u\to v\in A^e\cap A_{bt}} \lambda_{bt}^3 & \forall u\to v \in A^{\text{shu}} \\ c_{u,v}^e + \sum_{n,t:u\to v\in A^e\cap A_{nt}^1} \lambda_{nt}^1 & \forall u\to v \in A^{\text{dwe}} \cup A^{\text{ser}} \cup A^{\text{sto}} \\ c_{u,v}^e & \text{other} \end{cases} \tag{17}$$

subject to constraints (2) ~ (8), and (12).

When performing the dual iteration scheme, the Lagrangian multipliers λ_{nt}^1, λ_{nt}^2, λ_{bt}^3 are regarded as known parameters (Brännlund et al., 1998), and the modified arc cost $\bar{\bar{c}}_{u\to v}^e$ can be updated by Eq. (17).

The most conventional dual optimization technique, called subgradient method (Brännlund et al., 1998; Caprara et al., 2002; Zhang et al., 2019), is employed to update the Lagrangian multipliers λ_{nt}^1, λ_{nt}^2, λ_{bt}^3 in the $(k+1)$-th iteration, as expressed in Eq. (18)~(20). β_1^k, β_2^k, β_3^k are the step size parameters when updating.

$$\lambda_{nt}^{1,k+1} = \max\{0, \lambda_{nt}^{1,k} + \beta_1^k \cdot R_{nt}^1/(k+1)\}, \tag{18}$$

$$\lambda_{nt}^{2,k+1} = \max\{0, \lambda_{nt}^{2,k} + \beta_2^k \cdot R_{nt}^2/(k+1)\}, \tag{19}$$

$$\lambda_{bt}^{3,k+1} = \max\{0, \lambda_{bt}^{3,k} + \beta_3^k \cdot R_{bt}^3/(k+1)\}, \tag{20}$$

5.2 Problem decomposition based on ADMM

To further enhance the robustness and convexity of the LR framework, the Augmented Lagrangian Relaxation (ALR) method incorporates quadratic penalty terms of relaxed coupling constraints into the objective function. This method has been shown to exhibit superior performance compared to LR, as demonstrated in recent studies (Khodoomi & Sahebi, 2023; Song et al., 2023; Song & Cheng, 2022)

Despite the advantages of ALR, the introduction of quadratic penalty terms into the objective function

renders it nonlinear and makes the model difficult to decompose. However, the combination of ALR with the Block Coordinate Descent (BCD) method, specifically the Alternating Direction Method of Multipliers (ADMM), can address this issue by allowing for the sequential update of variables in a block-by-block manner (Bertsekas, 1997; Yao et al., 2019). As a result of the integration of ALR with ADMM, the relaxed ALR can be decomposed, thereby addressing the nonlinearity issue introduced by the quadratic penalty terms. Compared to LR, ADMM presents several advantages, including the ability to break the disconnection between subproblems of a single EMU, as demonstrated in recent studies (Zhang et al., 2019). This disconnection, also referred to as the symmetry deficiency of LR (Xue et al., 2023; Zhang et al., 2019), hinders the feasibility of the lower bound solution obtained through LR and makes it difficult to derive an upper bound. Therefore, ADMM is used for our problem.

To relax the incompatibility constraints (9) ~ (11) by ADMM, slack variables are introduced to convert these constraints from inequality to equality constraints, see Zhang et al. (2019), Q. Zhang et al. (2020), Feng et al. (2023) and Zhong et al. (2024). Let constraints (9) be the example and be equality constraint, then get $R^1_{nt} + s^1_{nt} = 0$ by introducing slack variable s^1_{nt}. Since$\sum_{e\in E}\sum_{u\to v\in A^e\cap A^1_{nt}} x^e_{u,v} \geq 0$, the range of the variable s^1_{nt} is $[0,1], \forall n \in N^2\backslash N^{\text{int-tra}} t \in T'_n$. Similarly, set slack variable $s^2_{nt} \in [0,1]\ \forall n \in N^1\backslash N^{\text{int-tra}}$, $t \in T'$; set slack variable $s^3_{bt} \in [0,1]\ \forall b \in B, t \in T'_b$.

After adding the quadratic terms into Obj. (13), the ADMM relaxation model $\mathbf{M}_{\text{ADMM}}$ is as follows:

$$\begin{aligned}&\min Z_{\text{ADMM}}(x,\lambda) = Z_{LR} + \frac{\rho_1}{2}\sum_{n\in N^2\backslash N^{\text{int-tra}}}\sum_{t\in T'}(R^1_{nt} + s^1_{nt})^2 + \\ &\frac{\rho_2}{2}\sum_{n\in N^1\backslash N^{\text{int-tra}}}\sum_{t\in T'}(R^2_{nt} + s^2_{nt})^2 + \frac{\rho_3}{2}\sum_{b\in B}\sum_{t\in T'}(R^3_{bt} + s^3_{bt})^2.\end{aligned} \tag{21}$$

subject to constraints (2) ~ (8), and (12).

The train-specific sub-model $\mathbf{M}^e_{\text{ADMM}}$ is expressed in Eq. (22), with the linearization process explained in Appendix A.

$$Z^e_{\text{ADMM}} = \sum_{e\in E}\sum_{u\to v\in A^e} c^e_{u\to v} x^e_{u\to v} + \sum_{u\to v\in A^{\text{shu}}} x^e_{u\to v} D^1_{u\to v} + \sum_{u\to v\in A^{\text{dwe}}\cup A^{\text{ser}}\cup A^{\text{sto}}} x^e_{u\to v} D^2_{u\to v} = \sum_{u\to v\in A} \bar{c}^e_{u\to v} x^e_{u\to v}\ . \tag{22}$$

subject to constraints (2) ~ (8), and (12).

where

$$\begin{aligned}&D^1_{u\to v} = \sum_{n,t:u\to v\in A^e\cap A^1_{nt}}\left\{\lambda^1_{nt} + \max\left[0, \frac{\rho_1}{2}\left(2\left(\sum_{e'\in E\backslash e}\sum_{a''\in A^1_{nt}} x^{e'}_{a''}\right) - 1\right)\right]\right\} + \\ &\sum_{n,t:u\to v\in A^e\cap A^2_{nt}}\left\{\lambda^2_{nt} + \max\left[0, \frac{\rho_2}{2}\left(2\left(\sum_{e'\in E\backslash e}\sum_{a''\in A^2_{nt}} x^{e'}_{a''}\right) - 1\right)\right]\right\} + \\ &\sum_{b,t:u\to v\in A^e\cap A_{bt}}\left\{\lambda^3_{bt} + \max\left[0, \frac{\rho_3}{2}\left(2\left(\sum_{e'\in E\backslash e}\sum_{a''\in A_{bt}} x^{e'}_{a''}\right) - 1\right)\right]\right\}\ \ \forall u\to v \in A^{\text{shu}};\end{aligned} \tag{23}$$

$$D^2_{u\to v} = \sum_{n,t:u\to v\in A^e\cap A^1_{nt}}\left\{\lambda^1_{nt} + \max\left[0, \frac{\rho_1}{2}\left(2\left(\sum_{e'\in E\backslash e}\sum_{a''\in A^1_{nt}} x^{e'}_{a''}\right) - 1\right)\right]\right\}\ \ \forall u\to v \in A^{\text{dwe}}\cup A^{\text{ser}}\cup A^{\text{sto}}. \tag{24}$$

The Lagrangian multipliers λ^1_{nt}, λ^2_{nt}, λ^3_{bt} in ADMM are also updated by Eq. (18) ~ (20). However, the step sizes $\frac{\beta^k_1}{k+1}, \frac{\beta^k_2}{k+1}, \frac{\beta^k_3}{k+1}$ in Eq. (18) ~ (20) are set to be the quadratic coefficients ρ_1, ρ_2, ρ_3 respectively, in accordance with the standard approach outlined in Boyd (2011) and Zhang et al. (2019).

5.3 Solution procedure

The following Fig.10 represents the framework of the entire approach. It should be noted that there is no rigorous mathematical proof that the solution of ADMM can be used as the lower bound of the problem, as mentioned in Song and Cheng (2022). However, there is a rigorous mathematical proof that the solution

of pure LR can be used as the lower bound of the problem. For a comparison of the characteristics of the three dual approaches, namely pure LR, ALR with the method of multipliers, and ADMM, we refer to Table 1 in Song and Cheng (2022). In our solution framework, pure LR will be used to obtain the lower bound by using the dual iteration scheme in Brännlund et al. (1998), while ADMM and an upper bound heuristic (detailed in Section 5.5) will be used to obtain the upper bound. The upper bound is the objective function value of the best solution found.

The framework of ADMM is similar with Zhang et al. (2019) and Yao et al. (2019). The key difference lies in designing problem-specific shortest path search algorithms and an upper bound heuristic. In this paper, a modified label setting algorithm in Section 5.4 is specifically designed to handle the service times constraints (5) ~ (8), and a pre-assignment strategy is devised to expedite the problem-solving process. The upper bound heuristic also incorporates an inner loop local search for further optimization.

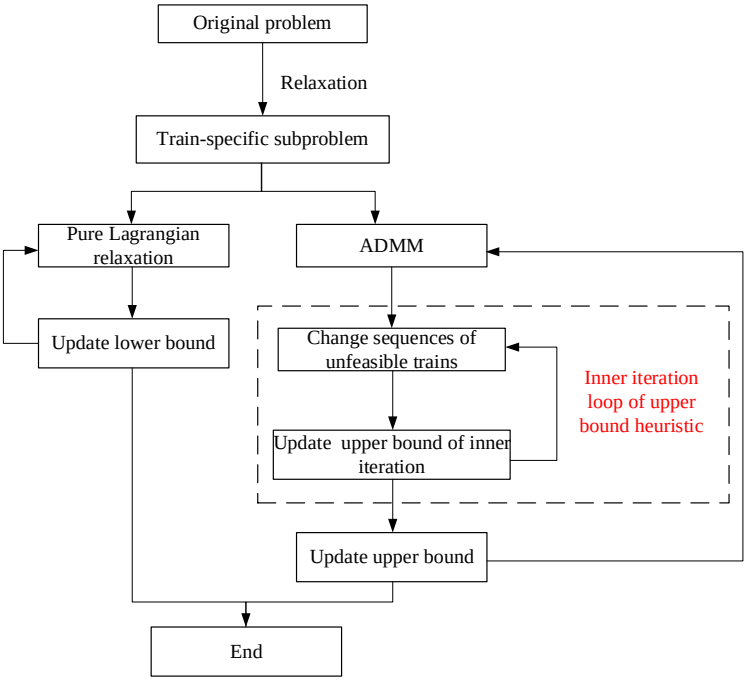


Figure 10: The framework of the whole approach

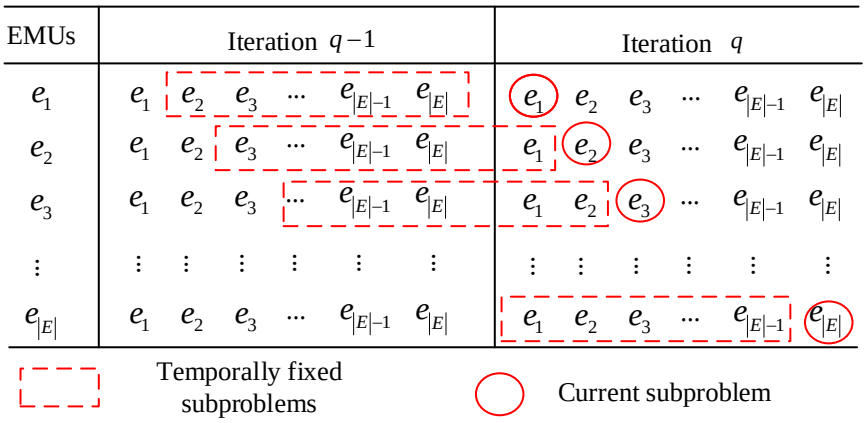


Figure 11: Illustration of the sequential iteration process in the ADMM-based approach.

A complete ADMM-based solution procedure is shown in Algorithm 1. The solution procedure of Algorithm 1 consists of five steps. In the initialization process in Step 1, the iteration number k, Lagrangian multipliers $\lambda_a^{1,k}, \lambda_a^{2,k}, \lambda_a^{3,k}$, penalty parameters $\rho_1^k, \rho_2^k, \rho_3^k$, the upper bound and lower bound solutions $\{X_{\mathrm{UB}}^0\}$ and $\{X_{\mathrm{LB}}^0\}$, the best lower/upper bound value LB^*/ UB^* are provided. In step 2, the associated single-EMU

plan is solved sequentially using the modified label-setting algorithm outlined in Appendix B. Similar to the Block Coordinate Descent method, the sequential iteration process is depicted in Fig.11. During solving the EMU e plan at iteration k, only the time-space least-cost path for EMU e is optimized, while the time-space paths for other EMUs remain temporarily fixed. The key element in breaking the symmetry is updating the dynamic resource usage cost by Eq. (22) at each inner iteration. Unlike LR, parallel computing cannot be employed in the ADMM procedure since the solution of previous sub-models can impact the current sub-model. Furthermore, the values of multipliers in ADMM are updated by Eq. (18) ~ (20), with the step-sizes equal to $\rho_1^k, \rho_2^k, \rho_3^k$. In Step 3, the best upper solution by the upper bound heuristic is generated in Section 5.5, which uses the feasible part of the solution by ADMM in Step 2. In Step 4, the best lower bound is obtained by using a pure LR dual iteration scheme. The Lagrangian multipliers in pure LR are updated using Eq. (18) ~ (20). Ultimately, Step 5 determines whether to continue the iterative algorithmic process by returning to Step 2 or to terminate the algorithm upon reaching the maximum iteration generation.

Algorithm 1. an ADMM-based solution procedure

//**Step 1**: Initialization
Initialize $k = 0$; $\lambda_a^{1,k}, \lambda_a^{2,k}, \lambda_a^{3,k}, \rho_1^k, \rho_2^k, \rho_3^k$, $\{X_{\mathrm{UB}}^0\}$ and $\{X_{\mathrm{LB}}^0\}$, $\mathrm{LB}^* = -\infty$, $\mathrm{UB}^* = +\infty$;
//Step 2: Solve the Model $\mathbf{M}_{\mathrm{ADMM}}^e$ for each EMU e sequentially
//Step 2.1: Using the label setting algorithm for each EMU e
For each EMU $e \in E$
Update the modified arc cost $\bar{c}_{u\to v}^e$ by Eq. (22);
Find the space-time shortest path for EMU e using the modified label setting algorithm in Appendix B;
End for
// **Step 2.2**: Update the Lagrangian multipliers in ADMM
Update the Lagrangian multipliers in ADMM by Eq. (18) ~ (20), with the step-sizes are equal to $\rho_1^k, \rho_2^k, \rho_3^k$.
//Step 3: Generate the upper bound solution and compute UB^k
//**Step 3.1:** Find upper bound solution $\{X_{\mathrm{UB}}^k\}$ for the original Model **M**
Based on the solution generated by Step 2.1, generate the best upper solution $\{X_{\mathrm{UB}}^k\}$ by the upper bound heuristic in Section 5.4.
// **Step 3.2:** Compute UB^k
Compute UB^k by substituting solution $\{X_{\mathrm{UB}}^k\}$ in Eq. (1);
$\mathrm{UB}^* = \min\{\mathrm{UB}^*, \mathrm{UB}^k\}$.
// **Step 4:** Generate the lower bound solution and compute LB^k
// **Step 4.1:** Solve the pure LR Model $\mathbf{M}_{\mathrm{LR}}^e$ to get $\{X_{\mathrm{LB}}^k\}$
For each EMU $e \in E$
Update the modified arc cost $\tilde{c}_{u,v}^e$ by Eq. (17);
Solve sub-model $\mathbf{M}_{\mathrm{LR}}^e$ in parallel by the modified label setting algorithm in Appendix B.
End for
// **Step 4.2:** Compute LB^k
Compute LB^k by substituting solution $\{X_{\mathrm{LB}}^k\}$ in Eq. (14);
$\mathrm{LB}^* = \max\{\mathrm{LB}^*, \mathrm{LB}^k\}$.
// **Step 4.3:** Update the Lagrangian multipliers in pure LR
Update the Lagrangian multipliers in pure LR by Eq. (18) ~ (20).
//Step 5: Termination condition test
Compute the relative gap by Gap= $\frac{\mathrm{UB}^*-\mathrm{LB}^*}{\mathrm{LB}^*}\times 100\%$:
If the terminal condition is satisfied, output LB^* and UB^*; otherwise, k:=k+1 and go back to Step 2.

5.4 The local-search-based upper bound heuristic

In the context of minimizing operating costs, the upper bound refers to the best (lowest) operating cost of a solution found so far. An upper bound solution is feasible if there are no cancelled EMUs, and near-feasible if some EMUs are cancelled. Then, penalties for cancelled EMUs are added to the operating cost. In the results of the Model $\mathrm{M}_{\mathrm{ADMM}}^e$ at Step 2.1 of Algorithm 1, if EMU e doesn't have any conflicts with other EMUs after checking constraints (5) ~ (8), then the solution of EMU e will be directly used as a part of the upper bound solution. $|\mathrm{E_F}|$ is the number of conflict-free EMUs in the solution produced by ADMM in each iteration. The common approach is to solve the plan for the remaining $|\mathrm{E}| - |\mathrm{E_F}|$ EMUs in sequence. After extensive testing, solving the remaining EMUs only once is likely to fail to obtain a feasible solution, thus wasting the solutions generated by the current ADMM iteration. Since one ADMM iteration takes a long time, making full use of the solutions of the ADMM iteration to conduct multiple feasible solution searches

for the remaining EMUs can improve the searching speed and solution quality. Therefore, this paper improves the common approach and introduces an inner loop to perform local search when solving the plan for the remaining EMUs. A total of NumLS sequences of the remaining $|E| - |E_F|$ EMUs will be generated as neighbourhoods to perform a local search in one upper bound iteration, where NumLS=min $\{5, 2^{|E_F|-1}\}$. Based on numerical testing, employing the inner loop for solving yields a speed improvement of more than three times in obtaining feasible solutions for the first time. Additionally, the frequency of EMU service cancellations is significantly reduced compared to not using the inner loop. In each inner iteration of different random sequences, the modified label setting algorithm in Appendix B is performed to get the feasible solutions of each remaining EMU. The time-space arcs occupied by the current EMU and conflict-free EMUs in the solution produced by ADMM are marked as unavailable for subsequent EMUs, see a similar handling approach in Carey and Carville (2003) and Xu and Dessouky (2022). Parallel computing cannot be used in the upper bound heuristic. After each inner iteration, the upper bound will be updated. The best solution of the inner iteration will be the upper bound of the outer iteration.

5.5 Track pre-assignment strategy

In general, large railway hubs have a significant number of tracks, especially platform tracks and storage tracks. The high number of homogeneous tracks results in a large solution space for each train, which affects the solving speed. To reduce the solution space, a track pre-assignment strategy will be applied in both the ADMM and upper bound heuristic. In the track pre-assignment strategy, for each iteration, each train is randomly assigned or manually assigned with a set of candidate tracks, such as two platform tracks and two storage tracks. The candidate track set of each train can be regarded as the L-list in Billionnet (2003). A similar approach is also used in Haahr et al. (2017) and Kamenga (2020), referred to as "Type and Track Decomposition". Then a certain train unit type is restricted to park on a select subset of tracks. If assigned randomly, for each platform and storage track, the number of trains that include that track in their set of candidate tracks cannot exceed a maximum limit. If there is a small number of trains, and the resources of the station and EMU depot are abundant, the number of candidate tracks can be set to a lower value. However, if there is a large number of trains, setting a smaller number of candidate tracks may exclude feasible or even optimal solutions and result in train cancellations. Therefore, the number of candidate tracks for each train needs to be determined according to the number of trains and tracks. On the other hand, if candidate tracks are determined based on human experience or station regulations, it is necessary to refer to the habitual uses of tracks for each direction at the station and the platform preference, such as assigning the same platform for certain pairs of trains for passengers to transfer. The station regulations include dividing the resources of the station into several yards based on directions or train speeds. Trains are only allowed to stop in their corresponding yards. For instance, at Beijing South Station, one of China's largest high-speed railway stations with 13 platforms and 24 tracks, the station is divided into three yards based on speed and direction. From north to south, they are the relatively-low-speed yard (3 platforms, 5 tracks), the Beijing-Shanghai direction high-speed yard (6 platforms, 12 tracks), and the Beijing-Tianjin intercity yard (4 platforms, 7 tracks) (Guo et al., 2020). Similar partitioning regulations also exist in some large EMU depots, especially in the storage yards.

Under the application of the track pre-assignment strategy, after a round of upper bound inner loop iterations, the solutions of the feasible trains are kept unchangeable, and the unfeasible train will be recalculated. In the recalculation, the available tracks will not be limited to the previous set of candidate tracks, that is, all tracks are available for unfeasible trains to choose from. This "remedial measure" helps to repair the defect of track pre-assignment strategy cutting feasible solution spaces.

6 Case Study

This section includes a small case study based on a virtual hub, as well as a large real-world case study based on the Jinan West hub (shown in Fig. C.1) in China, to test our solution approach. The scale of a real-world hub is much larger than that of the small artificial generated hub. The ADMM approach was implemented in C# and tested on a desktop workstation with a 2.3GHz 12-core processor and 128GB of memory. The input and output datasets, C# codes, for these cases can be downloaded from

https://data.mendeley.com/datasets/ nzdyy6vyf4. A TPSSS planning software based on the method in this paper is also developed. The interface of the software is shown in Fig. C.2. In all cases, the time unit is set to 1 min. The standard time t_{ep}^{ser} of cleaning, repair and wheel turning for long EMUs is 20, 150, 180 minutes respectively, while for short EMUs, it is 15, 80, and 120 minutes respectively. Due to the complex layout of the hub, the duration of a shunting arc may vary. For example, the time it takes for a train to stop on a shunting route close to the main line or far away from the main line is different and needs to be determined based on the actual situation at the station. Generally, it is 1~3 minutes. For example, the time from BJ1 to PT3 in Fig. C.2 is 1 minute. The safety headway time is set to 2 min. For the cost of the arcs, set $c_{ep}^{\mathrm{ser}} = 3, c_{ep}^{\mathrm{dwe}} = 3, c_{e}^{\mathrm{dwesto}} = 2$ for 8-railcar EMUs, and set $c_{ep}^{\mathrm{ser}} = 6$, $c_{ep}^{\mathrm{dwe}} = 6$, $c_{e}^{dwesto} = 4$ for 16-railcar EMUs. Set $c^{\mathrm{fix}} = 100$, $c_{ea}^{\mathrm{shu}} = 10$. Actually, more headway time leads to a more robust plan, but results in a smaller solution space for parking and routing. The maximum time duration pre_d_e that is allowed for early entry into the station is set to 25. The maintenance time window for the station infrastructure is set from 0:00 to 4:00. During this interval, no EMUs are allowed to park on the platform tracks. The input data of hub infrastructure and EMUs for both the virtual hub and the real-world hub are described in detail in the corresponding subsections. In this section, unless otherwise specified, all instances are conducted under the progressive strategy and the pre-assignment strategy. Under the pre-assignment strategy, each EMU is assigned 2~3 candidate tracks for platform parking and storage, respectively.

6.1 A small virtual hub

This section's computational study is centered on a small virtual hub depicted in Fig. 3. The planning horizon spans 720 time units (12 hours). To test the algorithm's performance, 9 small-scale test instances are generated. In all instances, β_1^k, β_2^k, β_3^k=1, ρ_1, ρ_2, ρ_3=35. The number of EMUs $|\mathrm{E}|$ is set to be 16, 18 and 20. In each instance, there are 10 network trains, that is, the number of depot trains is set to be 6, 8 and 10. All instances involve three types of services: cleaning, repair, and wheel turning. The maximum computation time is limited to 1 hour, and the maximum number of iterations is set to 200. As the shortest path search for the lower bound can be computed in parallel and takes relatively little time, only the lower bound is calculated during the first 100 iterations to warm up the multipliers (Yao et al., 2019). The ADMM and upper-bound heuristic is executed during the latter 100 iterations, and Table 3 presents the computational results for these instances. The "$|\mathrm{E_p}|$" column displays the number of EMUs with 2 and 3 services, respectively, in two sub-columns. To account for the small virtual hub's single wheel turning track, the maximum number of EMUs allowed to undergo wheel turning is restricted to 2. The "$|\mathrm{E_{DEP}}|$" column reports the number of depot trains, including long EMUs and short EMUs, in the "$|\mathrm{E_L}|$" and "$|\mathrm{E_S}|$" sub-columns. The "$|\mathrm{E_{cc}}|$" column indicates the number of cancelled EMUs whose parking or movements were rejected due to capacity limitations. The "LB" and "UB" columns represent the best lower bound and best upper bound, while the "GAP" column shows the best gap between the lower and upper bounds, GAP=(UB-LB) / LB×100%. For the instance with $|\mathrm{E_{cc}}| > 0$, the GAP value is not presented, as the GAP just depends on the given penalty parameter. The "IB" column denotes the iteration number at which the best upper bound was obtained. The "CT" column specifies the computation time in CPU seconds, and the "Av.CT" column denotes the average computation time in CPU seconds for one iteration. Since the EMUs' conflict-free schedule in the ADMM solution directly serves as part of the upper bound solution, the quality of the ADMM solutions is measured using the feasible rate FEA=Avg.$|\mathrm{E_F}|/|\mathrm{E}| \times 100\%$, as shown in the "FEA" column, where Avg.$|\mathrm{E_F}|$ is the mean number of conflict-free EMUs in the solution produced by ADMM in each iteration.

Table 3: Computational results of the small virtual hub with various EMU scales.

Inst.	$\lvert E\rvert$	$\lvert E_p\rvert$		$\lvert E_{DEP}\rvert$		Solution statistics							
		2	3	$\lvert E_L\rvert$	$\lvert E_S\rvert$	$\lvert E_{cc}\rvert$	LB	UB	GAP	IB	CT	Av.CT	FEA

S-1	16	4	2	1	5	0	18694	18698	0.02%	116	1857	13	95.63%
S-2	16	4	2	3	3	0	20811	21213	1.89%	112	1794	12	94.94%
S-3	16	4	2	5	1	0	22598	22983	1.67%	117	1803	13	91.25%
S-4	18	6	2	2	6	0	24404	24781	1.52%	195	3032	24	84.50%
S-5	18	6	2	4	4	0	26483	27258	2.84%	189	3048	24	85.17%
S-6	18	6	2	6	2	1	28846	32939	-	150	3600	38	75.25%
S-7	20	8	2	3	7	0	30144	31667	4.81%	143	3600	37	79.81%
S-8	20	8	2	5	5	2	31992	36446	-	158	3600	48	68.48%
S-9	20	8	2	7	3	3	35828	42546	-	143	3600	48	67.26%

Table 3 shows that an increase in the number of EMUs leads to a larger solution gap and longer computation time. The cancellation of EMUs only occurs in instances S-6, S-8, and S-9. There are no conflicts among the remaining conflict-free EMUs in instances S-6, S-8, and S-9. Thus, dispatchers can improve the computer-generated plan further with their human experience. This will save most of the time of dispatchers, compared to starting a manual planning from scratch (van den Broek et al., 2022). Instances without cancelled trains reveal that the FEA values are generally high, indicating that ADMM can produce high-quality and almost-feasible solutions when the EMU count falls within the hub capacity. As a result, the upper bound heuristic for non-feasible EMUs in ADMM is invoked less frequently. In most instances, the best upper bound is generated in the early iterations, suggesting that the solution produced by ADMM is of relatively high quality, despite its infeasibility.

When the number of long and short EMUs is relatively balanced, the small virtual hub can accommodate up to 8 EMUs for cleaning, repair, or wheel turning tasks. This approach can assist decision-makers in determining hub capacity during strategic decision-making processes. In instances where the number of EMUs is equal, such as S-1, S-2, and S-3, the UB and GAP tends to increase as the percentage of long EMUs increases. The higher cost for long EMUs results in the increasing UB and also affects the GAP, as long EMUs occupy more resources.

Fig.12 displays the Gantt chart visualization of the best feasible solution for instance S-7. Each row in the chart represents a track, and the length of the rectangle represents the track's occupation period. The number in the rectangle is the index of EMU. Thick rectangles represent long EMUs, while thin rectangles represent short EMUs. The upper half of the row indicates Position 1, while the lower half represents Position 2. The relocation strategy enables EMU-20, planned after EMU-12, to use the limited available resources to park and wait for service resources to become available. Among the hub, EMU-9 requires the minimum number of shunting operations, which was 2, since it doesn't need any service. Regarding the saw-move as two shunting operations, EMU-16 requires the maximum number of shunting operations, which is 9. Dispatchers have the option to adjust the fixed penalty cost of shunting, which can result in more or fewer shunting operations depending on the number of drivers. For EMU-13, EMU-14, EMU-17, and EMU-19, additional storage is available between two services. This flexible storage strategy prevents the EMUs from becoming blocked on the track of the first service. However, Xu and Dessouky (2022) only consider two services and do not allow for additional storage operations between two services. Saw-moves are performed on the clean track C1 for EMU-17 and EMU-19, as well as during the relocation processes of EMU-16 and EMU-20.

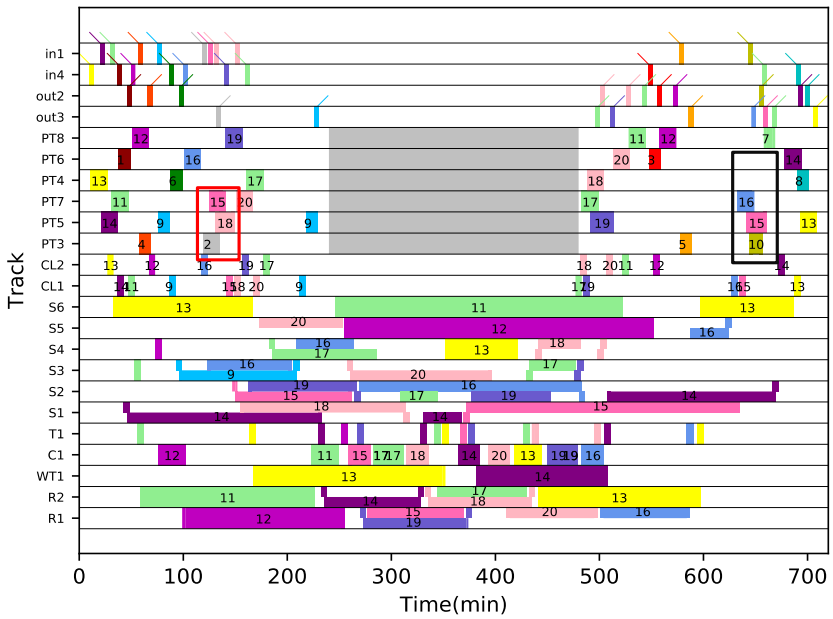


Figure 12: The schedule of the feasible upper bound solution of the small virtual hub instance S-7

6.2 Real-world case study

Fig. C.1 in Appendix C illustrates the layout of Jinan West railway hub, which consists of Jinan West Station and Jinan Depot. Currently, Jinan West Station connects trains from four directions: Beijing (BJ), Shanghai (SH), Shijiazhuang (SJZ), and Qingdao (QD). Jinan West Station is an important station along the Beijing-Shanghai high-speed railway, the busiest and most profitable high-speed railway line in China. In addition, a new high-speed railway line connecting Zhengzhou will be opened at Jinan West hub by the end of 2023. Jinan West station has 2 main lines and 15 platform tracks. Adjacent to Jinan West Station, there is Jinan EMU Depot, which has 6 repair tracks (R1~R6), 2 cleaning tracks (C1, C2), 3 wheel turning tracks (WT1 ~WT3), 24 storage tracks (S1 ~S24). There are 37 critical switches, indicated as black dots in Fig. C.1.

Real-world EMU data from the hub are collected to test the performance of the algorithm. Station information was obtained from https://www.12306.cn and http://www.lltskb.com/. In all instances, β_1^k, β_2^k, β_3^k=1, ρ_1, ρ_2, ρ_3=35. For the planning horizon from 20:00 to 10:00 (+1 day), there are 86 EMUs at the hub. Terminal EMUs and the following day's originating EMUs belonging to the same rolling stock are considered as a single EMU. There are 66 network trains, including 65 passing EMUs with stops and 1 immediately reverse EMU. There are 20 depot trains, including 15 EMUs needing service, 3 EMUs needing night parking, and 2 non-immediately reverse EMUs. The maximum computation time is set to 1 hour, which seems too long, but it's acceptable for generating a plan for hubs and EMUs of this scale (van den Broek et al., 2022; Xu & Dessouky, 2022). In practice, for such a large-scale hub, a feasible shunting and servicing plan by the dispatcher takes on average 4-6 hours, and then it is still almost impossible to measure the quality of the plan. The solution result is shown as real-world instance L-1 in Table 4. Considering the future growth in passenger flow at the Jinan West hub, we conducted additional evaluations of the algorithm's performance by simulating increased numbers of EMUs and varying ratios of long and short EMUs. Table 4 presents a summary of the input statistics and computational results for these instances. The arrival and departure time are determined by the fixed macroscopic timetable, so it is irrational to randomly generate input information for network trains in Jinan West Hub. Therefore, instances L-2 ~ L-9 are based on the real-world instance L-1 with additional 10 and 20 depot trains, respectively. The input information of additional depot trains is randomly generated by a standard generator. For example, the arrival time and departure time are randomly generated from a discrete uniform distribution between the early stage and the late stage of the planning horizon, as also did in Xu and Dessouky (2022). The result shows that the gap values are all within 5%. Although L-9 cancels more trains than L-8 and has a lower FEA, the average

computation time per iteration is shorter than that of L-8. This is because L-9 has more long EMUs, and the EMUs at the end of the schedule queue have fewer available resources to choose from. As a result, these EMUs spend less time searching for the shortest time-space path during the search.

The Gantt chart visualization of the best feasible upper bound solution for the instance L-8 are displayed in Appendix C. It can be found from Fig. C.3 that the bottleneck of Jinan West Hub's capacity is in the connection lines and repair tracks. Other types of tracks, including platform tracks, have some idle capacity. The possible subsequent transformation and upgrading of hub infrastructure should focus on adding more connection lines and repair tracks. Some EMUs, such as EMU-79, EMU-85, EMU-95, EMU-100, and EMU-105, stay at the station for longer than the standard stop time due to congested connection lines and related switch sections. This indicates that through joint optimization, the time for trains to enter and leave the depot can be flexibly determined based on the occupancy of the connection lines and switches. This aligns with our motivation (2) mentioned in Section 1.

From Fig. C.3, it is evident that the utilization of service tracks remains relatively uniform throughout the entire planning horizon. This is due to the consistent cost assigned to the physical arcs throughout the planning horizon. In practice, certain depots may prioritize early completion of train servicing. In such cases, it is possible to assign different costs to arcs at the beginning and end of the planning horizon. For instance, during the initial phase of the planning horizon, the cost of service arcs could be set at half of the original cost to encourage early train servicing. In summary, the costs of arcs can be tailored according to the specific circumstances at hand.

Due to the incapability of Cplex / Gurobi to find feasible solutions within the specified time, a local search (LS) algorithm is also employed for comparison with the ADMM-based heuristic. To ensure fairness in the comparison, the track pre-assignment strategy is also incorporated into the LS algorithm. In the LS algorithm, during each iteration, EMUs are sequentially solved according to a predetermined order, utilizing the modified label setting algorithm as described in Section 5.4. After each iteration, a local search is performed on the train order, including changing the candidate tracks of EMUs, swapping the orders of conflicting trains and random changing the order of trains arriving or departing within a short time period. The next iteration proceeds based on the updated train order. The local search is also used in Jacobsen and Pisinger (2011) and van den Broek et al. (2022), despite some differences in the generation of neighborhoods. The maximum solving time of LS is set to 1 hour, and the results are presented in the "LS.Obj" column of Table 4. Comparing these results with the "UB" column demonstrates that the ADMM-based heuristic proposed in this paper outperforms the local search algorithm.

Table 4: Computational results of Jinan West hub with various EMU scales.

Inst.	$\|E\|$	$\|E_p\|$		$\|E_{DEP}\|$		Solution statistics							
		2	3	$\|E_L\|$	$\|E_S\|$	$\|E_{cc}\|$	LB	UB	GAP	IB	Av.CT	FEA	LS.Obj
L-1	86	11	4	4	16	0	54366	54892	0.96%	26	190	95.61%	56685
L-2	86	11	4	10	10	0	59098	59471	0.63%	27	176	95.70%	61280
L-3	86	11	4	16	4	0	66103	66846	1.11%	23	138	95.98%	68194
L-4	96	15	6	8	22	0	72070	73308	1.69%	28	324	93.65%	79429
L-5	96	15	6	15	15	0	77159	78720	1.98%	31	251	92.15%	84086
L-6	96	15	6	22	8	0	84520	86470	2.25%	22	285	91.86%	98007
L-7	106	17	8	12	28	0	92581	96245	3.81%	22	380	88.66%	103046
L-8	106	17	8	20	20	0	101284	104511	3.09%	26	453	89.02%	115401
L-9	106	17	8	28	12	4	110056	121331	-	23	427	84.95%	139875

6.3 Sensitivity analysis

In this section, a sensitivity analysis of inputs and different strategies is performed.

6.3.1 Effect of different ratios of the capacities of the station to the depot.

This sub-section tests the solution under different capacity configurations of the hub. Two sets of test cases, based on the small virtual hub, are designed, one with 18 and the other with 20 EMUs. Each set of cases has 5 instances with a different track resource configuration. $|H_{PLA}|$, $|H_{STO}|$, $|H_{REP}|$ respectively represent the number of platform tracks at the station, the number of storage tracks at the EMU depot, and the number of repair tracks at the EMU depot. The total number of the three types of tracks is 14. The results are shown in Table 5, where the "Av.PT" column represents the average parking time in minutes at platform tracks of the best solution without cancelling EMUs. It can be observed that as the number of platform tracks increases (the number of tracks at the EMU depot decreases), the dwell time of EMUs at the station also increases. This indicates that when the resources at the EMU depot are insufficient, the algorithm is able to guide EMUs to stay longer on the platform tracks at the station, thereby improving the coordination of track resources in the hub. To some extent, our motivation (3) mentioned in Section 1 has been realized. For instance, when station resources are tight, some terminating trains may enter the depot early, while departing trains may leave the depot later. When depot resources are tight, some terminating trains may enter the depot later, while departing trains may leave the depot earlier, and non-immediately reverse trains do not enter the depot for storage. It can also be seen from Table 5 that with the same number of platform tracks, an increase in the number of repair tracks results in less average parking time at the station. This is due to a sufficient number of repair tracks improving the smoothness of service switching, reducing the number of storage and shunting operations. Moreover, a higher number of repair tracks leads to better solution quality, as reflected in lower GAP values, lower Av.CT values, and higher FEA values.

Table 5: Computational results of the small virtual hub with different ratios of the capacities of the station to the depot.

Inst.	$\|E\|$	$\|H_{PLA}\|$	$\|H_{STO}\|$	$\|H_{REP}\|$	$\|E_{cc}\|$	LB	UB	GAP	Av.CT	FEA	Av.PT
C18-1	18	4	8	2	0	23098	23769	2.82%	29	85.87%	22
C18-2	18	4	7	3	0	23009	23567	2.37%	25	90.28%	22
C18-3	18	6	6	2	0	23064	23311	1.06%	25	88.22%	39
C18-4	18	6	5	3	0	23317	23383	0.28%	21	92.67%	30
C18-5	18	7	5	2	0	23427	23556	0.55%	23	89.39%	35
C20-1	20	4	8	2	0	28479	29882	4.69%	45	78.33%	25
C20-2	20	4	7	3	0	28108	28850	2.57%	37	84.80%	25
C20-3	20	6	6	2	0	28475	29225	2.57%	41	79.71%	38
C20-4	20	6	5	3	0	28476	29153	2.32%	27	89.05%	32
C20-5	20	7	5	2	1	28598	30859	-	40	79.08%	-

6.3.2 Comparison between the conservative strategy and the progressive strategy

In this section, the conservative strategy and the progressive strategy are compared. First, the small instances S-1, S-4 and S-7 are selected for comparison. Then, real data of the platforming plan for the

network trains in large Instance L-1 is collected as the fixed plan under the conservative strategy.

For each small instance, network trains are randomly given platform tracks for 5 times to generate 5 sub-cases and simulate the fixed schedule of the network trains under the conservative strategy. Then, mark the time-space resources occupied by these network trains as unavailable, the remaining depot EMUs' schedule are generated by the algorithm. For each instance, we calculate the average of all results for each sub-case under the conservative strategy, denoted as instances S-1-C, S-4-C and S-7-C shown in Table 6. The columns "#.$|E_{cc}|$", "#.UB", and "#.Av.CT" represent the increased number of cancelled EMUs, the increase in upper bound, and the increase in the average computation time in CPU seconds for one iteration, respectively, compared to the progressive strategy. The "Num.Cross" and "#.Num.Cross" column represent the number of times all EMUs cross the main line and the number of increased times all EMUs cross the main line compared to the progressive strategy. Under the progressive strategy, the number of times for crossing the main line in S-1, S-4, and S-7 is 3, which increases under the conservative strategy. Additionally, under the conservative strategy, if the network train schedule is fixed and cannot be changed, some depot trains may be affected and cancelled. This will result in the dispatchers modifying the platform plan for network trains. However, under the progressive strategy, no EMUs are cancelled for small instances S-1, S-4, and S-7. Regarding the motivation (1) mentioned in Section 1, the progressive strategy avoids modifications to the train platforming plan of network trains at the station.

Fig. 13 shows the Gantt chart for the best solution of a sub-case of S-7-C under the conservative strategy. The network trains EMU-2 and EMU-15 are randomly assigned on platform track PT5. In the red rectangle, depot train EMU-18 cannot park at PT3 because PT3 is not connected to the depot, according to the layout in Fig.3. Therefore, EMU-18 is cancelled. However, in the red rectangle in Fig. 12 under the progressive strategy, network train EMU-2 is placed on PT3 by the algorithm, and EMU-18 is successfully assigned. In the black rectangle in Fig. 13, due to platform track PT7 being occupied by network train EMU-10, depot train EMU-16 has to cross the main line when departing from platform track PT6, although it can depart successfully. However, in the black rectangle in Fig. 12 under the progressive strategy, network train EMU-10 is placed on PT3 by the algorithm, and EMU-16 can depart from PT7 without crossing the main line. Therefore, such kind of crossing can be avoided. The cost of the shunting arc for crossing the main line is 5 times the cost of a regular shunting arc, i.e., 100. This value can be adjusted based on the busyness of the station and the dispatcher's preferences.

Therefore, the progressive strategy can schedule all trains in the hub comprehensively within a similar solving time and improve the quality of the depot plan, avoiding the interference of the fixed network train schedule under the conservative strategy.

Table 6: Computational results of the small virtual hub under the conservative strategy

Inst.	$\|E_{cc}\|$	#.$\|E_{cc}\|$	UB	#.UB	Av.CT	#.Av.CT	Num.Cross	#Num.Cross
S-1-C	0.8	0.8	20690.00	1992.00	15.05	2.01	3.8	0.8
S-4-C	0.4	0.4	25545.80	764.80	23.85	-0.07	3.8	0.8
S-7-C	0.8	0.8	32334.80	667.80	34.15	-2.70	3.6	0.6

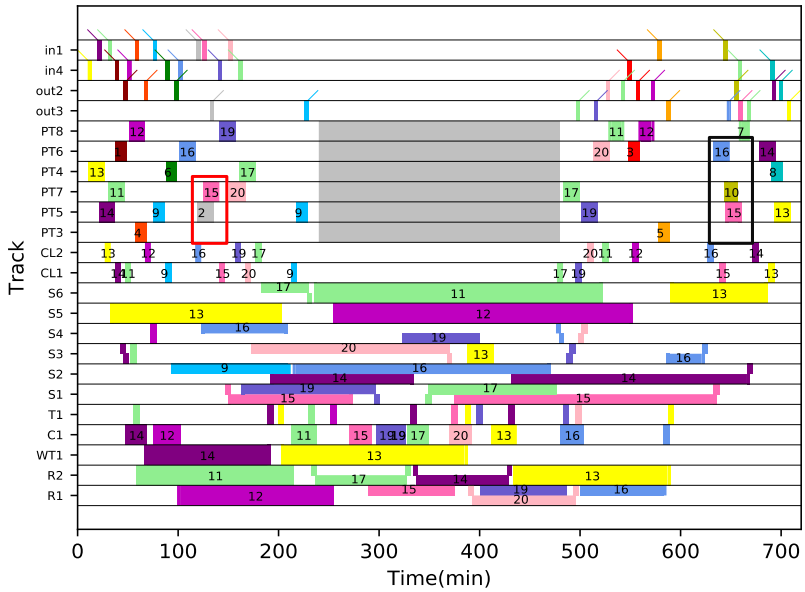


Figure 13: The schedule of the best upper bound solution of the small virtual hub instance S-7-C

For the large instance L-1, the real platforming plan of 66 network trains is also collected from https://www.12306.cn. Under the conservative strategy, the instance is denoted as L-1-C. The platforming plan for network trains is kept fixed, and the solution method is applied to the remaining depot trains. The results under the same parameter conditions as instance L-1 are shown in Table 7. Compared to the progressive strategy where no trains are cancelled in instance L-1, two depot trains are cancelled in L-1-C because their routes are blocked by the network trains. Due to the high penalty for cancelled trains, the best upper bound of L-1-C has increased. From the column "#.Av.CT", it can be observed that despite the decrease in the total number of trains to be planned, the computation time has increased. This is due to two reasons. Firstly, network trains usually have short station stops, and the time for searching the shortest time-space path can be almost negligible. Secondly, due to the occupation of resources by network trains, more depot trains are cancelled in each iteration, resulting in more trains undergoing the upper bound heuristic.

Table 7: Computational results of the real-world instance L-1 of the Jinan West hub under the conservative strategy

Inst.	$\|E_{cc}\|$	$\#.\|E_{cc}\|$	UB	#.UB	Av.CT	#.Av.CT	Num.Cross	#Num.Cross
L-1-C	2	2	48720	3928	325	135	7	0

6.3.3 Comparison between the pre-assignment and non-pre-assignment strategy

In theory, pre-assignment can absolutely reduce the solution space. As mentioned in the experiment in Kamenga (2020), reducing the number of available shunting tracks helped to reduce the number of related variables by 65% to 75%. In this section, we will test the solution quality of the instances in Sections 6.1 and Section 6.2 under the non-pre-assignment strategy (instances with the "NPA" suffix, such as "S-1-NPA"), in order to determine whether pre-assignment leads to worse solution quality. Under the non-pre-assignment strategy, the results for small and large instances are shown in Table 8 and Table 9, respectively. Columns starting with "#." indicate the increase in corresponding statistical results compared to those under the pre-assignment strategy. In most cases, although the solution space is reduced, applying the pre-assignment strategy leads to better solution quality. This is because the pre-assignment strategy avoids a large number of symmetric solutions, which results in faster solving speed and, within the computation time limit, not-worse or even better solving performance. The effectiveness of the pre-assignment strategy is clearly more

prominent for instances with a greater number of EMUs. However, the choice of which strategy to adopt should be determined based on the real circumstances and rules.

Table 8: Computational results of the small virtual hub under the non-pre-assignment strategy

Inst.	|E|	$\|E_{cc}\|$	#.$\|E_{cc}\|$	UB	#.UB	GAP	#.GAP	Av.CT	#.Av.CT
S-1-NPA	16	0	0	22884	-99	1.25%	-0.42%	16	4
S-2-NPA	16	0	0	21213	0	1.90%	0.01%	15	2
S-3-NPA	16	0	0	19164	466	2.37%	2.35%	15	2
S-4-NPA	18	1	0	32962	23	12.71%	0.28%	42	4
S-5-NPA	18	0	0	27605	347	4.06%	1.22%	23	-1
S-6-NPA	18	0	0	24788	7	1.54%	0.02%	26	2
S-7-NPA	20	4	1	43530	984	-	-	57	9
S-8-NPA	20	3	1	37588	1142	-	-	51	3
S-9-NPA	20	0	0	32129	462	6.19%	1.38%	42	5

Table 9: Computational results of the Jinan West hub under the non-pre-assignment strategy

Inst.	|E|	$\|E_{cc}\|$	#.$\|E_{cc}\|$	UB	#.UB	GAP	#.GAP	Av.CT	#.Av.CT
L-1-NPA	86	0	0	54661	-231	1.48%	0.52%	240	50
L-2-NPA	86	0	0	59475	4	1.32%	0.69%	191	16
L-3-NPA	86	0	0	66309	-537	0.90%	-0.21%	176	38
L-4-NPA	96	1	1	75238	1930	-	-	810	486
L-5-NPA	96	1	1	79427	707	-	-	791	539
L-6-NPA	96	2	2	91960	5490	-	-	692	407
L-7-NPA	106	5	5	101334	5089	-	-	850	470
L-8-NPA	106	7	7	114752	10241	-	-	820	367
L-9-NPA	106	5	1	123406	2075	-	-	729	302

6.3.4 Effect of different step size parameters

The Small Instances S-4 and S-7 are used to test the effect of different step size parameters β_1^k, β_2^k, β_3^k and different quadratic coefficients ρ_1, ρ_2, ρ_3. First, different β_1^k, β_2^k, β_3^k are tested in two instances and the iteration profiles of the lower bound are shown in Fig.14. The step size parameters are reflected in the ascending step-like iteration profile, and the step is obviously larger when the step parameters β_1^k, β_2^k, β_3^k are selected to be larger. In particular, when β_1^k, β_2^k, β_3^k are equal to 0.2, the iteration profiles are almost smooth

curves with no obvious steps, which may result in a local optimum and lead to a lower objective value. When $\beta_1^k, \beta_2^k, \beta_3^k$ are equal to 2, in the early stage of the iteration process, the lower bound do not change, the iterated profiles do not even show a tendency to converge in 200 iterations. When $\beta_1^k, \beta_2^k, \beta_3^k$ equal to 0.2, 0.5 and 1, the objective values increase immediately in the early stage, and it shows convergence in the 200th generation. So, $\beta_1^k, \beta_2^k, \beta_3^k$ equal to 0.2, 0.5 and 1 are suitable for the small instance. Considering the computational time requirement of the problem, $\beta_1^k, \beta_2^k, \beta_3^k$ equal to 1 with the best lower bound is chosen as the final input parameter of small instances.

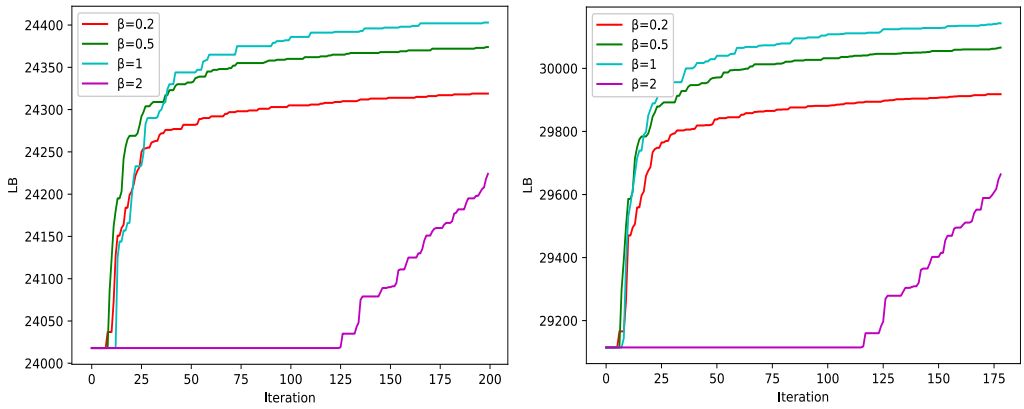


Figure 14: Iteration profiles of the lower bound of Small Instances S-4 and S-7with different $\beta_1^k, \beta_2^k, \beta_3^k$: (a) Small Instance S-4, (b) Small Instance S-7.

6.3.5 Effect of different quadratic coefficients

Other papers (Yao et al., 2019; Q. Zhang et al., 2020) use relatively small quadratic coefficients ρ. This paper uses a quantitative approach to determine the appropriate quadratic coefficients ρ for this problem. The feasible rate FEA will be used to measure the quality of ADMM solutions under different values of quadratic coefficients ρ. The quality of the ADMM solutions for Small Instances S-4 and S-7 under different values of quadratic coefficients ρ is tested. The values of FEA are shown in Fig.15. If quadratic coefficients ρ are too small, the ability to overcome the symmetry deficiency is poor, and it cannot output a high-quality ADMM solution, i.e., there are still a large number of EMUs in the ADMM solution that conflict with other EMUs; if quadratic coefficients ρ are too large, the step sizes in the ADMM iteration process are too large to obtain convergence. Taken together, the quadratic coefficients ρ_1, ρ_2, ρ_3 are all chose equal to 35 as the quadratic penalty parameter in ADMM for the small instance.

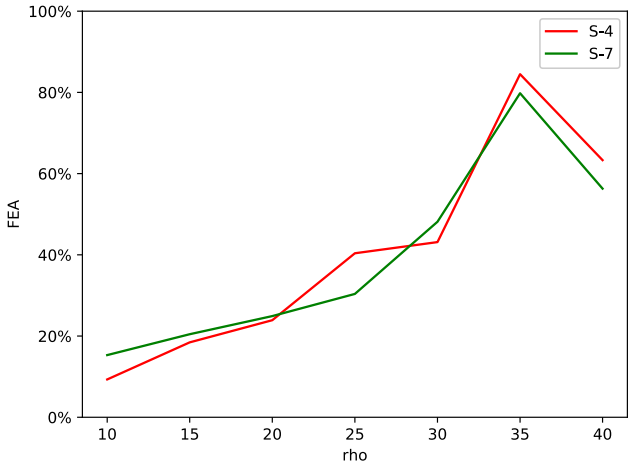


Figure 15: Values of FEA under different ρ_1, ρ_2, ρ_3

7 Conclusion

This study presents a multi-commodity network flow model based on a new two-layer time-space network for addressing the TPSSS problem at railway hubs in China. An ADMM-based heuristic method with an iterative framework that includes inner and outer double loops is developed to solve the model. A small case study and a large real-world case study are investigated to verify the validity of our proposed model. Based on the results of the case studies, several conclusions can be drawn. These are as follows:

(1) The proposed ADMM-based algorithm can provide high-quality solutions for the integrated problem of high-speed train platforming and depot scheduling, if there are no cancelled EMUs. The approach is applicable to different types of hub configurations and allows for trains to have varying numbers of services and flexible storages.

(2) By setting different costs for arcs, the algorithm can automatically adjust the total station and depot dwell time, resulting in a more balanced utilization of resources throughout the hub.

(3) Compared to the conservative strategy of fixing the network trains' platforming plan, the progressive strategy reduces the probability of depot train cancellations and reduces the use of unreasonable routes, such as crossing the main line, without significantly increasing the solution time.

(4) By applying the pre-assignment strategy and using the "remedial measure", the solving speed can be increased, especially in large-scale cases. In most cases, the solution quality is better than that of the non-pre-assignment strategy, especially when there are a large number of trains.

(5) A sensitivity analysis is conducted to discover the effects of important parameters of the approach. The step size parameters of LR and quadratic coefficients in ADMM can greatly affect the results and efficiency of the approach, and need to be determined based on the problem size.

Future research will focus on several aspects. First, crew scheduling, including maintenance workers and drivers) at the hub also will be jointly considered (Kamenga et al., 2021; van den Broek, 2022). Second, the method in this paper is suitable for designing a basic plan, but for real-time scheduling cases such as late trains or overtime repair (Kamenga et al., 2021; Shi et al., 2024), it is imperative to design more efficient algorithms. Third, the case study exclusively examines the scenario where a hub consists of a single station and a single depot. In reality, larger hubs may involve multiple stations connected to a single depot, or a single station connected to multiple depots. While our network exhibits flexibility for expansion, the algorithm's performance in such complex configurations necessitates further validation.

Reference

Artigues, C., Bourreau, E., Jost, V., Kedad-Sidhoum, S., & Ramond, F. (2018). Trains do not vanish: the ROADEF/EURO challenge 2014. *Annals of Operations Research*, *271*(2), 1091-1105. https://doi.org/10.1007/s10479-018-3066-x

Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society*, *48*(3), 334-334.

Billionnet, A. (2003). Using integer programming to solve the train-platforming problem. *Transportation Science*, *37*(2), 213-222. https://doi.org/10.1287/trsc.37.2.213.15250

Bosi, T., Bigi, F., D'Ariano, A., Viti, F., & Pineda-Jaramillo, J. (2024). Optimal management of full train load services in the shunting yard: A comprehensive study on Shunt-In Shunt-Out policies. *Computers & Industrial Engineering*, *188*, 109865. https://doi.org/https://doi.org/10.1016/j.cie.2023.109865

Boyd, S. P. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

Brännlund, U., Lindberg, P. O., Nou, A., & Nilsson, J.-E. (1998). Railway timetabling using Lagrangian relaxation. *Transportation Science*, *32*(4), 358-369. https://doi.org/10.1287/trsc.32.4.358

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

Caprara, A., Galli, L., & Toth, P. (2011). Solution of the Train Platforming Problem. *Transportation Science*, *45*(2), 246-257. https://doi.org/10.1287/trsc.1100.0366

Carey, M., & Carville, S. (2003). Scheduling and platforming trains at busy complex stations. *Transportation Research Part a-Policy and Practice*, *37*(3), 195-224. https://doi.org/10.1016/S0965-8564(02)00012-5

Corman, F., Goverde, R. M. P., & D'Ariano, A. (2009). Rescheduling Dense Train Traffic over Complex Station Interlocking Areas. *Robust and Online Large-Scale Optimization*, *5868*, 369-386. https://doi.org/10.1007/978-3-642-05465-5

Feng, T., Lusby, R. M., Zhang, Y., Peng, Q., Shang, P., & Tao, S. (2023). An ADMM-based dual decomposition mechanism for integrating crew scheduling and rostering in an urban rail transit line. *Transportation Research Part C: Emerging Technologies*, *149*, 104081. https://doi.org/https://doi.org/10.1016/j.trc.2023.104081

Freling, R., Lentink, R. M., Kroon, L. G., & Huisman, D. (2005). Shunting of passenger train units in a railway station. *Transportation Science*, *39*(2), 261-272. https://doi.org/10.1287/trsc.1030.0076

Guo, B., Wu, W., & Zhu, X. (2020). Analysis and model the train set shunting plan safety optimization problem in the hub type high-speed railway station. 2020 International Conference on Urban Engineering and Management Science (ICUEMS), Zhuhai, China.

Haahr, J. T., Lusby, R. M., & Wagenaar, J. C. (2017). Optimization methods for the train unit shunting problem. *European Journal of Operational Research*, *262*(3), 981-995. https://doi.org/10.1016/j.ejor.2017.03.068

He, M., Tang, Q., Gupta, J. N., Yin, D., & Zhang, Z. (2022). The shunting scheduling of EMU first-level maintenance in a stub-end depot. *Flexible Services and Manufacturing Journal*, 1-43. https://doi.org/10.1007/s10696-022-09459-6

He, M., Tang, Q., Gupta, J. N. D., Zhang, Z., & Cao, J. (2023). Robust scheduling of EMU first-level maintenance in a stub-end depot under stochastic uncertainties. *Engineering Applications of Artificial Intelligence*, *123*, 106398. https://doi.org/10.1016/j.engappai.2023.106398

Jacobsen, P. M., & Pisinger, D. (2011). Train shunting at a workshop area. *Flexible Services and Manufacturing Journal*, *23*(2), 156-180. https://doi.org/10.1007/s10696-011-9096-1

Kamenga, F. (2020). *Combinatorial optimization for integrating rolling stock management and railway trac scheduling in passenger stations* [Doctoral dissertation, Université de Lille]. https://hal.science/tel-03169424v1

Kamenga, F., Pellegrini, P., Rodriguez, J., & Merabet, B. (2021). Solution algorithms for the generalized train unit shunting problem. *Euro Journal on Transportation and Logistics*, *10*. https://doi.org/10.1016/j.ejtl.2021.100042

Kavička, A., & Krýže, P. (2021). Dynamic Automated Search of Shunting Routes within Mesoscopic Rail-Traffic Simulators. *Journal of Advanced Transportation*, *2021*, 8840516. https://doi.org/10.1155/2021/8840516

Khodoomi, M., & Sahebi, H. (2023). Robust optimization and pricing of Peer-to-Peer energy trading considering battery storage. *Computers & Industrial Engineering*, *179*, 109210. https://doi.org/10.1016/j.cie.2023.109210

Kroon, L. G., Lentink, R. M., & Schrijver, A. (2008). Shunting of passenger train units: an integrated approach. *Transportation Science*, *42*(4), 436-449. https://doi.org/10.1287/trsc.1080.0243

Li, H., Jin, M., He, S., Ye, Z., & Song, J. (2017). Optimal track utilization in electric multiple unit maintenance depots. *Computers & Industrial Engineering*, *108*, 81-87. https://doi.org/10.1016/j.cie.2017.03.031

Li, Z., Yin, J., Chai, S., Tang, T., & Yang, L. (2023). Optimization of system resilience in urban rail systems: Train rescheduling considering congestions of stations. *Computers & Industrial Engineering*, *185*, 109657. https://doi.org/https://doi.org/10.1016/j.cie.2023.109657

Lin, B., & Zhao, Y. (2021). Synchronized optimization of EMU train assignment and second-level preventive maintenance scheduling. *Reliability Engineering & System Safety*, *215*, 107893. https://doi.org/https://doi.org/10.1016/j.ress.2021.107893

Lu, G. Y., Ning, J., Liu, X. B., & Nie, Y. (2022). Train platforming and rescheduling with flexible interlocking mechanisms: An aggregate approach. *Transportation Research Part E-Logistics and Transportation Review*, *159*. https://doi.org/10.1016/j.tre.2022.102622

Lusby, R., Larsen, J., Ehrgott, M., & Ryan, D. (2011a). Railway track allocation: models and methods. *OR Spectrum*, *33*(4), 843-883. https://doi.org/10.1007/s00291-009-0189-0

Lusby, R., Larsen, J., Ryan, D., & Ehrgott, M. (2011b). Routing Trains Through Railway Junctions: A New Set-Packing Approach. *Transportation Science*, *45*(2), 228-245. https://doi.org/10.1287/trsc.1100.0362

Lusby, R. M., Larsen, J., Ehrgott, M., & Ryan, D. M. (2013). A set packing inspired method for real-time junction train routing. *Computers & Operations Research*, *40*(3), 713-724. https://doi.org/10.1016/j.cor.2011.12.004

Mahmoudi, M., & Zhou, X. S. (2016). Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state-space-time network representations. *Transportation Research Part B-Methodological*, *89*, 19-42. https://doi.org/10.1016/j.trb.2016.03.009

Padovano, A., Longo, F., Manca, L., & Grugni, R. (2024). Improving safety management in railway stations through a simulation-based digital twin approach. *Computers & Industrial Engineering*, *187*, 109839. https://doi.org/https://doi.org/10.1016/j.cie.2023.109839

Riezebos, J., & van Wezel, W. (2009). k-Shortest routing of trains on shunting yards. *OR Spectrum*, *31*(4), 745-758. https://doi.org/10.1007/s00291-008-0140-9

Sels, P., Vansteenwegen, P., Dewilde, T., Cattrysse, D., Waquet, B., & Joubert, A. (2014). The train platforming problem: The infrastructure management company perspective. *Transportation Research Part B-Methodological*, *61*, 55-72. https://doi.org/10.1016/j.trb.2014.01.004

Shi, J., & Li, H. (2019). Optimization of the Shunting Operation Plan at Electric Multiple Units Depots. *Journal of Advanced Transportation*, *2019*. https://doi.org/10.1155/2019/9120239

Shi, J., Li, H., Wang, D., & Vansteenwegen, P. (2024). Reschedule the shunting and service scheduling plan after disturbances at a Chinese high-speed railway depot. *Ieee Access*. https://doi.org/10.1109/ACCESS.2024.3368807

Song, M., Cheng, L., & Zhang, Y. (2023). Joint location optimization of charging stations and segments in the space-time-electricity network: An augmented Lagrangian relaxation and ADMM-based decomposition scheme. *Computers & Industrial Engineering*, *183*, 109517. https://doi.org/10.1016/j.cie.2023.109517

Song, M. C., & Cheng, L. (2022). Solving the reliability-oriented generalized assignment problem by Lagrangian relaxation and Alternating Direction Method of Multipliers. *Expert Systems with Applications*, *205*. https://doi.org/10.1016/j.eswa.2022.117644

Tang, Z., Jiang, Y., & Yang, F. (2023). An Efficient Lagrangian Relaxation Algorithm for the Shared Parking Problem. *Computers & Industrial Engineering*, *176*, 108860. https://doi.org/https://doi.org/10.1016/j.cie.2022.108860

van den Broek, R., Hoogeveen, H., van den Akker, M., & Huisman, B. (2022). A local search algorithm for train unit shunting with service scheduling. *Transportation Science*, *56*(1), 141-161. https://doi.org/10.1287/trsc.2021.1090

van den Broek, R. W. (2016). *Train Shunting and Service Scheduling: an integrated local search approach* [Master dissertation, Utrecht University]. https://studenttheses.uu.nl/handle/20.500.12932/24118

van den Broek, R. W. (2022). *Towards a Robust Planning of Train Shunting and Servicing* [Doctoral dissertation, Utrecht University]. https://dspace.library.uu.nl/handle/1874/420470

Wang, J., Zhao, Y., Gronalt, M., Lin, B., & Wu, J. (2021). Synchronized optimization for service scheduling, train parking and routing at high-speed rail maintenance depot. *Ieee Transactions on Intelligent Transportation Systems*, *23*(5), 4525-4540. https://doi.org/10.1109/TITS.2020.3045852

Wang, Y., Gao, Y., Yu, X., Hansen, I. A., & Miao, J. (2019). Optimization models for high-speed train unit routing problems. *Computers & Industrial Engineering*, *127*, 1273-1281. https://doi.org/https://doi.org/10.1016/j.cie.2018.02.039

Wang, Y., Zhang, Z. M., Huisman, D., D'Ariano, A., & Zhang, J. C. (2022). A Lagrangian relaxation approach based on a time-space-state network for railway crew scheduling. *Computers & Industrial Engineering*, *172*. https://doi.org/10.1016/j.cie.2022.108509

Xu, X., & Dessouky, M. M. (2022). Train shunting with service scheduling in a high-speed railway depot. *Transportation Research Part C: Emerging Technologies*, *143*, 103819. https://doi.org/10.1016/j.trc.2022.103819

Xue, F., Zhang, X., Hu, P., Ma, X., & Chen, C. (2023). Metro crew planning with heterogeneous duty paths and period-cycle pattern considerations. *Computers & Industrial Engineering*, *182*, 109354. https://doi.org/https://doi.org/10.1016/j.cie.2023.109354

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

Zhang, B., Zhang, Y., D'Ariano, A., Bosi, T., Lu, G., & Peng, Q. (2023). Optimal platforming, routing, and scheduling of trains and locomotives in a rail passenger station yard. *Transportation Research Part C: Emerging Technologies*, *152*, 104160. https://doi.org/10.1016/j.trc.2023.104160

Zhang, Q., Lusby, R. M., Shang, P., & Zhu, X. (2020). Simultaneously re-optimizing timetables and platform schedules under planned track maintenance for a high-speed railway network. *Transportation Research Part C: Emerging Technologies*, *121*, 102823. https://doi.org/10.1016/j.trc.2020.102823

Zhang, Q., Lusby, R. M., Shang, P., & Zhu, X. (2022). A heuristic approach to integrate train timetabling, platforming, and railway network maintenance scheduling decisions. *Transportation Research Part B-Methodological*, *158*, 210-238. https://doi.org/10.1016/j.trb.2022.02.002

Zhang, Y. X., Peng, Q. Y., Yao, Y., Zhang, X., & Zhou, X. S. (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. X., Zhong, Q. W., Yin, Y., Yan, X., & Peng, Q. Y. (2020). A Fast Approach for Reoptimization of Railway Train Platforming in Case of Train Delays. *Journal of Advanced Transportation*, *2020*. https://doi.org/10.1155/2020/5609524

Zhong, M., Yue, Y., Zhou, L., & Zhu, J. (2024). Parallel optimization method of train scheduling and shunting at complex high‐speed railway stations. *Computer‐Aided Civil and Infrastructure Engineering*, *39*(5), 731-755. https://doi.org/10.1111/mice.13077

Zhong, Q. W., Lusby, R. M., Larsen, J., Zhang, Y. X., & Peng, Q. Y. (2019). Rolling stock scheduling with maintenance requirements at the Chinese High-Speed Railway. *Transportation Research Part B-Methodological*, *126*, 24-44. https://doi.org/10.1016/j.trb.2019.05.013

Zwaneveld, P. J., Kroon, L. G., & van Hoesel, S. P. M. (2001). Routing trains through a railway station based on a node packing model. *European Journal of Operational Research*, *128*(1), 14-33. https://doi.org/Doi 10.1016/S0377-2217(00)00087-4

Appendix A

The following section illustrates the linearization process using the method proposed by Yao et al. (2019) and Q. Zhang et al. (2020), with constraints (9) as an example,

$$(R_{nt}^1 + s_{nt}^1)^2 = \left(\sum_{e\in E}\sum_{u\to v\in A^e\cap A_{it}^1} x_{u,v}^e - 1 + s_{nt}^1\right)^2 = \left(\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e + \sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'} - 1 + s_{nt}^1\right)^2 = \left(\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e\right)^2 + \left(\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'} - 1 + s_{nt}^1\right)^2 + 2\left(\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'} - 1 + s_{nt}^1\right)\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e. \quad (25)$$

In the above equation, the value of $\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e$ must be 0 or 1, which can be guaranteed in the dynamic programming of EMU e to find the shortest path, so $\left(\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e\right)^2 = \sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e$. So

$$(R_{nt}^1 + s_{nt}^1)^2 = \left(\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e\right)\left[2\left(\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'}\right) + 2s_{nt}^1 - 1\right] + \left(\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'} - 1 + s_{nt}^1\right)^2. \quad (26)$$

According to the block-by-block update discipline, the model can be decomposed to several subproblems for each EMU e where the variables associated with other EMUs are fixed. Consequently, $\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'}$ in Eq. (26) can be regarded as a constant.

To determine the value of s_{nt}^1, Eq. (26) can be regarded as the quadratic function with the variable s in Eq. (27)

$$F(s) = s^2 + 2s\left(\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e + \sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'} - 1\right) + \text{Constant}. \quad (27)$$

where $F(s)$ is optimal when $s^* = 1 - \sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e - \sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'}$.

Case 1: Constant $\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'} = 0$, which means that no other trains reach vertex (i, t). Then $s^* = 1 - \sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e$, $F^*(s) = F(s^*) = F\left(1 - \sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e\right)$=0.

Case 2: $\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'} \geq 1$, which means that there exist other trains reach vertex (i, t). Then $s^* = 1 - \sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e - \sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'}$, considering $s \in [0,1]$, then $s^* = 0$.

$$F^*(s) = F(s=0) = \left(\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e\right)\left[2\left(\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'}\right) - 1\right] + \text{Constant}$$

In conclusion,

$$(R_{nt}^1 + s_{nt}^1)^2 = \begin{cases} \left(\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e\right)\left[2\left(\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'}\right) - 1\right] + Constant, & \sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'} \geq 1, \\ 0, & \sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'} = 0 \end{cases} \quad (28)$$

Therefore, the quadratic term can be linearized to Eq. (29)

$$(R_{nt}^1 + s_{nt}^1)^2 = max\left\{0, \left(\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^e\right)\left[2\left(\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^1} x_{u,v}^{e'}\right) - 1\right] + \text{Constant}\right\}. \quad (29)$$

Similarly,

$$(R_{nt}^2 + s_{nt}^2)^2 = max\left\{0, \left(\sum_{u\to v\in A^e\cap A_{nt}^2} x_{u,v}^e\right)\left[2\left(\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{nt}^2} x_{u,v}^{e'}\right) - 1\right] + \text{Constant}\right\}, \quad (30)$$

$$(R_{bt}^3 + s_{bt}^3)^2 = max\{0, \left(\sum_{u\to v\in A^e\cap A_{bt}} x_{u,v}^e\right)[2\left(\sum_{e'\in E\setminus e}\sum_{u\to v\in A^e\cap A_{bt}} x_{u,v}^{e'}\right) - 1] + \text{Constant}\}. \quad (31)$$

Appendix B

The classical label setting algorithm is based on Dynamic Programming (DP) to solve the shortest path problem of an acyclic network with positive cost in Graph Theory. Regarding the cost of arcs as the distances, it is widely used in solving time-space least-cost path problems, see Yao et al. (2019) , Zhang et al. (2019), Q. Zhang et al. (2020), Feng et al. (2023) and Zhong et al. (2024). The detailed procedure is outlined in Algorithm 2.

Algorithm 2: Label setting algorithm

For each EMU e do
- Input starting vertex (n_e^a, a_e) with initial label $L(n_e^a, a_e) = [\Pi^{\text{ope}}, 0]$, $\Pi^{\text{ope}} = [0,0, \dots 0]$
- For each time instant $t \in [a_e, d_e]$ do
  - For each node n
    - For each existing label $[nn, tt, \Pi^{\text{ope}}(n,t), c(n,t)]$ of vertex (n,t) at time t, starting from node n search through the existing arc $a = (n,t) \to (m,t')$, where $t' = t + t_{nm}$:
      - Set $\Pi^{\text{ope-new}}(m,t') == \Pi^{\text{ope}}(n,t)$, then $\pi^p = \pi^p + 1$, if arc a is a service arc.
      - Set $c^{\text{new}}(m,t') == c(n,t) + c_{ea}$.
      - Create the new label $L^{\text{new}}(m,t') == [n,t,\Pi^{\text{ope-new}}(m,t'), c^{\text{new}}(m,t')]$.
      - If $\Pi^{\text{ope-new}}(m,t')$ with $\pi^p = 2$,
        - The new label $L^{\text{new}}(m,t')$ is derived.
      - If the current label $L^{\text{current}}(m,t') = (mm, tt, \Pi^{\text{ope}}(m,t'), c^{\text{current}})$ with $\Pi^{\text{ope}}(m,t') = \Pi^{\text{ope-new}}(m,t')$, and $c^{\text{current}} > c^{\text{new}}$.
        - Replace the current label $L^{\text{current}}(m,t')$ with the new label $L^{\text{new}}(m,t')$.
    - End;
  - End;
- End;

End;

Backtrace the path in the time-space network

For each EMU e, the label setting algorithm is employed to discover the least-cost path at the time-space network, where each time instant t functions as a stage for DP. Note that the pure label setting algorithm, where each vertex has only one label representing accumulated cost, can only be directly applied in a model that only has flow balance constraints. However, the constraints (5) ~ (8) aren't relaxed into the objective function. Therefore, Π^{ope} is introduced to the pure label setting algorithm to meet the constraints (5) ~ (8). Each vertex in the modified label setting may have multiple labels. One of the labels of the time-space vertex (m,t'), referred as $L(m,t') = [n,t,\Pi^{\text{ope}},c]$, has multiple dimensions, where node n is the node precedence, time instant t is the time precedence, Π^{ope} is the vector of completed times of all services, c is the accumulated cost. In Π^{ope}, $\Pi^{\text{ope}} = \left[\pi^1, \pi^2, \dots, \pi^{|P_e|}\right]$, where $\pi^p = \{0,1,2\}$, and $|P_e|$ represents the number of services for EMU e. In the initialization, let $\Pi^{\text{ope}} = [0,0, \dots 0]$. If EMU e traverses one service arc, let the corresponding service times indicator $\pi^p == \pi^p + 1$. When $\pi^p = 2$, the label is disqualified and will not be further explored. Since Π^{ope} will affect the cost of the path, in order to fairly compare the quality of two old and new labels of a vertex, it is necessary to follow that the two labels that have the same Π^{ope} can make the comparison. At this time, the label with the lower cost is better. The worse label will be

replaced and eliminated. If the label with the same Π^{ope} is not found, the new label will be retained as one of the labels of the vertex. Through the continuous setting and updating of labels, the least-cost time-space paths can be searched. The final label for each ending vertex (n_e^d, d_e) is determined by selecting the label that satisfies constraints (5) ~ (8). By backtracking from the final label, the least-cost path can be derived.

Appendix C

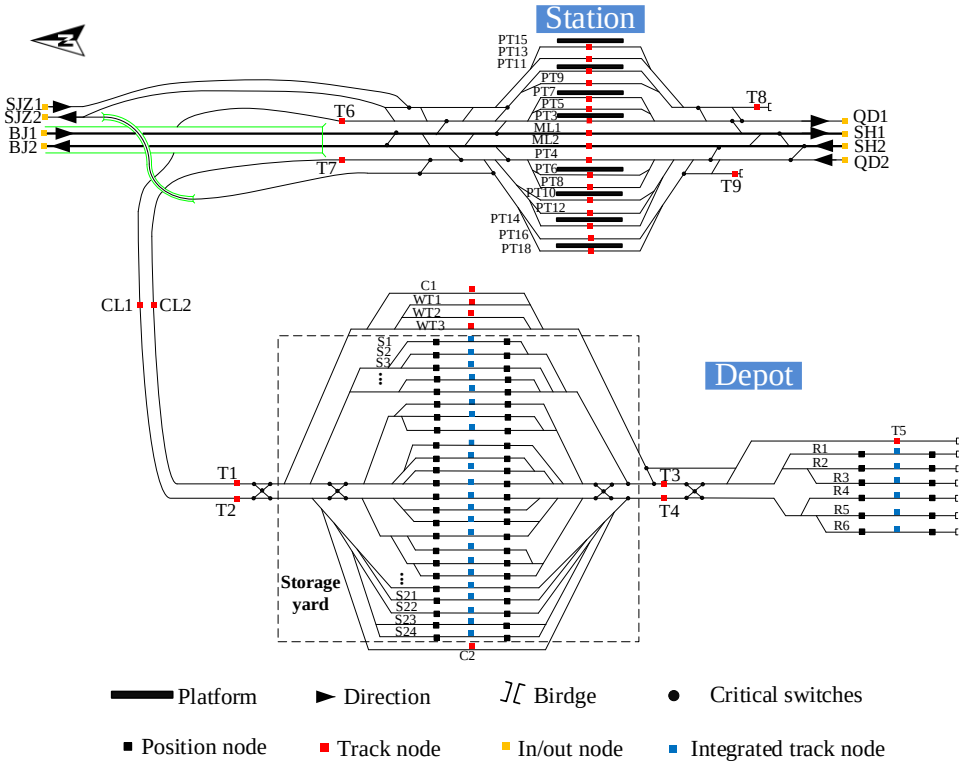


Figure C.1: The layout of Jinan West hub in China

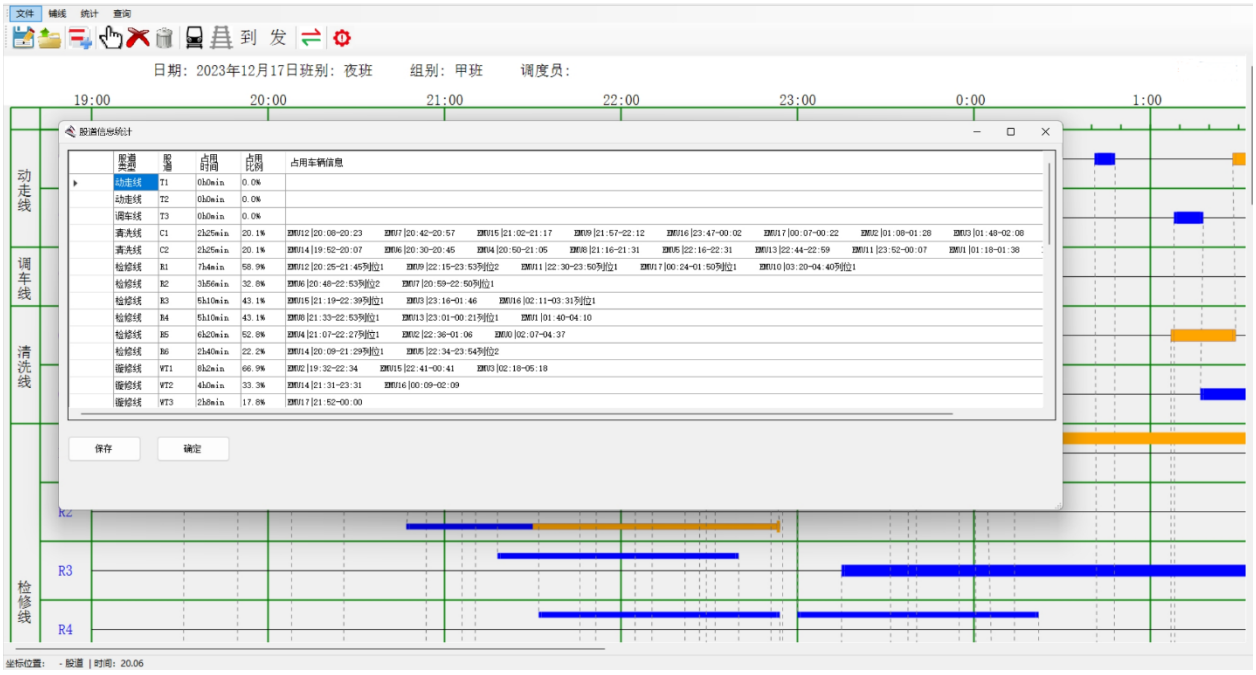


Figure C.2: The interface of the TPSSS planning software

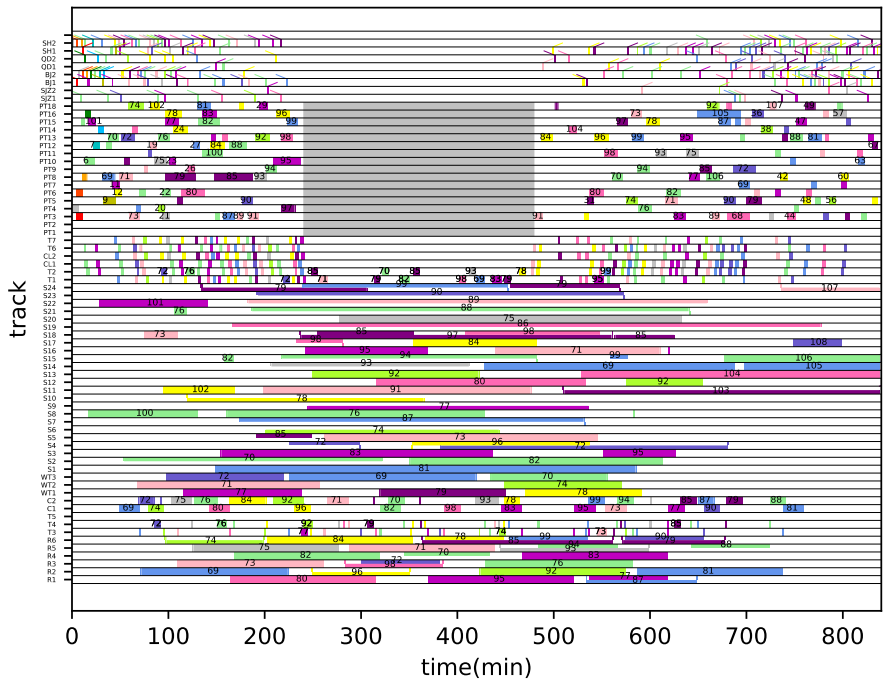


Figure C.3: The schedule of the feasible upper bound solution of the Jinan West hub instance L-8

Appendix D

In Shi and Li (2019), for solving the train shunting and service scheduling problem at the depot, the objective function only minimizes the total delay time of all EMUs, and the model only considers feasibility and ignores all costs. When total delay time =0, the plan is feasible.We calculate the results for each sub-case S-1, S-4 and S-7 using the method in Shi and Li (2019), denoted as instances S-1-DT, S-4-DT and S-7-DT shown in Table D.1. The columns "#.$|E_{cc}|$", "#Num.Shunt", "%Num.Shunt", and "#Num.Saw" represent the increased number of cancelled EMUs, the increased number of shunting movements, the percentage increase of shunting movements, and the increased number of saw-moves, respectively, compared to the results yielded by ADMM-based method. The column "#.CT" and "%.CT" represent the decreased value and percentage decrease of the computation time in CPU second, respectively. For the instance S-1, S-4 and S-7, the computation time is reduced by over 70%. However, the number of shunting movements is also increased by more than 40%, and unnecessary saw-moves occur, which will greatly increase the workload of the shunting drivers. For hubs with tight capabilities, only considering feasibility and ignoring all cost cannot obtain feasible solutions (no EMUs being cancelled) for the large-scale instances L-1~L-9 in Section 6.2. Therefore, although the solution time is increased by considering the comprehensive objective function and the cost of operations, the number of shunting movements and saw-moves can be greatly reduced by the proposed ADMM-based algorithm.

Table D.1: Computational results of the small virtual hub when only considering feasibility and ignore all cost

Inst.	#.$\|E_{cc}\|$	#Num.Shunt	%Num.Shunt	#Num.Saw	#.CT	%.CT
S-1-DT	0	10	43.47%	3	-1430	-77%
S-4-DT	0	21	61.76%	4	-2123	-70%
S-7-DT	1	25	62.50%	9	-2528	-70%