Efficient Real-Time Train Operation Algorithms
With Uncertain Passenger Demands

Jiateng Yin, Student Member, IEEE, Dewang Chen, Senior Member, IEEE,
Lixing Yang, Tao Tang, and Bin Ran

Abstract—The majority of existing studies in subway train operations focus on timetable optimization and vehicle tracking methods, which may be infeasible with disturbances in actual operations. To deal with uncertain passenger demands and realize real-time train operations (RTOs) satisfying multiobjectives, including overspeed protection, punctuality, riding comfort, and energy consumption, this paper proposes two RTO algorithms via expert knowledge and an online learning approach. The first RTO algorithm is developed by a knowledge-based system to ensure the multiple objectives with a constant timetable. Then, by considering uncertain passenger demand at each station and random running time errors, we convert the train operation problem into a Markov decision process with nondeterministic state transition probabilities in which the aim is to minimize the reward for both the total time delay and energy consumption in a subway line. After designing policy, reward, and transition probability, we develop an integrated train operation (ITO) algorithm based on Q-learning to realize RTOs with online adjusting the timetable. Finally, we present some numerical examples to test the proposed algorithms with real detected data in the Yizhuang Line of Beijing Subway. The results indicate that, taking the multiple objectives into account, the ITO algorithm outperforms both manual driving and automatic train operations. In addition, the ITO algorithm is capable of dealing with uncertain disturbances, keeping the total time delay within 2 s and reducing the energy consumption.

Index Terms—Real-time train operation, timetable, Q-learning, urban metro system, disturbances.

I. INTRODUCTION

THE demand for the modernization of transportation is getting urgent in China because of more and more serious transportation problems [1]. Recently, the urban rail transit has been regarded as one of the most important modes to release the city traffic congestion, due to its energy-saving, punctuality, safety and convenience.

As a core technology of a train control system, the subway train operation typically covers scheduling and real-time train operations [2]. For the planning process, a timetable is first generated, which specifies the running times and dwelling times of the trains in a railway system [3]. Based on this timetable, the railway control center is responsible for the trajectory planning with route information, such as line speed limits and gradients. Then, the speed profiles for the trains in railway system are derived, which aim to optimize some certain goals, e.g., energy consumption and riding comfort [4]. In real-time train operations, drivers receive the online information of trains (e.g., position, velocity) and operate the trains to track the predetermined speed profiles. More recently, automatic train operation (ATO) systems are applied to replace manual driving in some new subway lines, such as Yizhuang Line and Airport Line in Beijing Subway, Disneyland Resort Line in Hong Kong Subway, etc. The ATO focuses on designing control algorithms to track the speed profiles in real time. Indeed, the efficiency of an urban metro system is largely affected by the off-line optimized timetable and real-time train control methods [5], [6].

Nevertheless, the daily subway operations are inevitably disrupted by unexpected disturbances such as an unplanned stop, an arrival delay or a sudden change of passenger demands [8]. For example, according to the timetable in Yizhuang Line of Beijing Subway (YLBS), the predesigned dwelling time for a station is 30 s at Xiaocong station. If the passenger demand suddenly increases, the involved trains have to stay at this station 5 seconds longer for transferring passengers, which will typically cause a departure time delay. In such cases, the trains may arrive and depart later than the planned schedule [7], which will lead to the infeasibility of the timetable and even derive a secondary delay. For manual driving, the dispatchers can take actions to fix this problem manually by changing the timetable based on their experiences. However, it is difficult for ATO systems to find an optimal rescheduled timetable in time under disturbances, which may lead to a random time delay, reduction of efficiency or even cause certain trips to be cancelled. In addition, this problem seems to be more critical with the expanding of the metro system and increase of subway train’s density.

Accordingly, the real-time railway rescheduling (also called Train Timetable Rescheduling, TTR), which aims to regenerate a feasible timetable with the guaranteed quality, has been receiving tremendous attention in recent years [2], [6], [8].
However, existing studies typically consider timetable rescheduling as an optimization problem with a feasible decision model. As disturbances may occur at any time, it is difficult for ATO system to online adjust the timetable and derive a new speed profile. On the one hand, generating a high-quality rescheduled timetable within a short time is a challenging task because the real traffic modeling needs to consider a wide range of influencing factors with uncertainty properties. On the other hand, ATO algorithms that aim to track the off-line optimized trajectory are automatic and not intelligent. Once the timetable is changed, ATO needs to recalculate the trajectories of trains, which is not efficient in real-world applications. With these concerns, in Beijing Subway, the ATO mode has to be switched into manual driving mode in case of disturbances since existing algorithms are probably inconsistent with the changed timetables generated in various complex environments.

To address the train operation problems, with one train in a fixed segment, Yin et al. [9] emphasized the importance of intelligent control for urban metro systems and developed two train control algorithms based on an expert system and reinforcement learning. Nevertheless, the timetable is assumed to be constant in this study, which means that the driving process is not affected by any uncertain factor. Focusing on real-time train operations via a systematic approach, in this paper, we transform the train operation problem into a Markov decision process (MDP) with non-deterministic state transition probabilities considering the uncertain passenger demands at each station. Then, an integrated train operation (ITO) algorithm based on $Q$-learning, which minimizes the total time delay and energy consumption, is developed to obtain the optimal rescheduled timetable as well as the real-time driving strategies.

The rest of this paper is organized as follows. In Section II, we review some important literature on railway traffic management, involving single-train operation and timetable rescheduling. Section III describes the problem that exists in current train operation systems caused by disturbances. Section IV propose the real-time train operation algorithms based on an MDP model with non-deterministic state transition probabilities. In Section V, we describe the simulation platform by the filed data of Yizhuang Line in Beijing Subway (YLBS). Section VI presents numerical case studies to compare the proposed algorithms with manual driving and ATO in YLBS. We conclude this paper in Section VII.

II. LITERATURE REVIEW

Since railway traffic management has a great influence on the efficiency of urban metro networks, it has been studied for decades. Generally speaking, the existing research can be divided into two levels, i.e., individual-level which aims to realize optimal train operations for a single train in one segment, and system-level which concentrates on train timetabling and rescheduling. In what follows, we will review the literature on these two levels separately.

A. Individual Level

Single train operations determine the traction and braking controls, which have a direct impact on the performance of a subway system [5]. To obtain the energy-efficient train control strategy, Howlett and Pudney [10] and Cheng and Howlett [11] built a discrete control model and confirmed the fundamental optimality of the accelerate-coast-brake strategy based on the Pontryagin maximum principle. Liu and Golovitcher [12] established a continuous traction force model with constant efficiency to find the optimal control change points on a multilayer state-variable plain. Albrecht et al. [13] proved that the optimal switching points were uniquely determined for each steep section and the global optimal strategy was also unique. Su et al. [14] developed an iterative algorithm to calculate the optimal speed profile along the whole line. With the development of intelligent computing, the algorithms based on computational intelligence (e.g., genetic algorithm (GA), dynamic programming (DP) and ant colony optimization (ACO)) were regarded as effective ways to solve the NP-Hard problems [15]. Chang and Sim [16] applied a GA on a train control problem to generate an optimal coasting control based on the joint evaluation of punctuality, riding comfort and energy consumption. The performances of GA, ACO and DP were contrasted and analyzed by Lu et al. [17], which concluded that more than one method should be used to identify the optimal speed profile.

The other priority for the single train operation is automatic trajectory tracking methods [18]–[21]. For example, proportional-integral-derivative-based controllers are widely used in Beijing Subway train operations. Rigatos [18] proposed fuzzy stochastic automata of vehicle control. An iterative learning control based train trajectory tracking algorithms was developed in [19], which took into account the overspeed protection and safe headway control as well. To overcome the complexity and uncertainty of vehicle dynamics, including variable resistances, traction/braking saturation and in-train forces, more advanced control methods, such as fault-tolerant control [20], cruise control [22], [23] and robust adaptive control [21], were applied to tracking the speed profiles more precisely.

B. System Level

A railway timetable determines the arrival and departure times of a set of trains at each station [24], and a timetable is the precondition of subway train operations. The railway timetabling is regarded as an optimization problem with different objectives. For example, Higgins et al. [25] proposed a two-objective optimization model to minimize the fuel consumption and the delay of trains. To improve the utilization of regenerative braking energy, Yang et al. [26] derived a cooperative scheduling model to optimize the energy-saving timetable, and Dominguez et al. [27] designed a method to obtain the optimal speed profiles which minimize the net energy at substations. Dorfman and Medanic [28] presented a discrete event model and a greedy travel advanced strategy to improve the real-time quality. To reduce the delays for heterogeneous trains, Xu [29] et al. developed an efficient approach to schedule the trains with rapid computational speed. Besides, we refer to [30] for surveys of the literature on timetable planning.

Although the train operation problem has been widely studied in the past decades, the real-time train operation in subway
system is more or less unavoidably subject to unexpected disturbances. To deal with this problem, one feasible way is to design a robust timetable where buffer times are used to absorb a possible delay [31]. The other way is to adjust the arrival and departure times for trains in real-time which is called train timetable rescheduling. For example, in order to develop an online decision system for timetable rescheduling, Higgins et al. [32] formulated an integer programming model and developed several heuristic algorithms for the TTR problem, which included local search, genetic algorithm tabu search, and hybrid techniques. D’Ariano et al. [33] treated the TTR as a huge job shop scheduling problem with no store constraints and proposed a detailed alternative graph model for the train rescheduling problem, which utilized the accurate real-time information of train positions and speeds. A branch and bound algorithm was presented which included implication rules enabling to speed up the computation. Furthermore, they developed a real-time traffic management system, i.e. ROMA (Railway traffic Optimization by Means of Alternative graph) [34] to deal with the real-time disturbances by coordinating the speed of successive trains. Meng and Zhou [35] developed a single-line train rescheduling model where uncertain segment running times, recover times and the possibility of rescheduling decisions were considered. Aiming to minimize the total train delay time, they also designed a multi-layer branching algorithm under different stochastic scenarios. Yang et al. [36] proposed a fuzzy optimization approach for train rescheduling in a double-track railway network. A fuzzy variable-based recovery time was derived based on professional judgments or empirical estimates to handle the uncertain durations caused by accidents. Li et al. [37] considered the stochastic capacity recovery times of block tracks and proposed a new track-backup rescheduling (TBR) approach to obtain a conflict-free timetable which minimized the delay cost and the expected track changing cost by a mixed integer programming (MIP) model.

To the best of our knowledge, the majority of existing literature devotes to generating the rescheduled timetable with mathematical optimization models under the consideration of some specific disturbances. However in real-world applications, it is difficult to acquire a high-quality rescheduled solution within a short time using the optimization approaches [38]; besides, if multiple objectives such as punctuality, riding comfort, and energy consumption are considered in the calculation of the optimal speed profile, it will take a long time to put forward an altered speed profile in case of disturbances. Meanwhile, few study considers extra online information such as the amount of waiting passengers and dynamic characteristics of trains to predict the possible future states, e.g. running time prediction, expectations of entrance times and dwelling times [2].

In recent years, data-driven intelligent transportation systems (ITS) have emerged as an efficient way of improving the performance and enhancing service qualities as much more data are available to lead to a revolution in transportation development [39]. From this point, in this study we emphasize the importance of subway data and propose two real-time train operation algorithms via an online learning approach to deal with uncertain passenger demands in each station and improve the efficiency of a subway line as well. First, we focus on the optimal driving strategy under fixed running time between adjacent stations, and then, we combine the real-time train operation with online adjusting timetable in train operation process to recover from uncertain disturbances.

### III. Problem Statement

#### A. Definition of Symbols

For better understanding of this paper, we define the necessary symbols of the parameters of subway train operation process (see Fig. 1) as follows.

**Offline data by the timetable:**

- \( N \): number of stations;
- \( T_{di} \): departure time of the train at station \( i, i \in [1, N] \);
- \( T_{ai} \): arrival time of the train at station \( i, i \in [1, N] \);
- \( T_t \): total trip time;

**Online data:**

- \((s, v)\): position and speed of the train;
- \( t \): running time index;
- \( S_i \): distance between station \( i \) and station \( i+1 \);
- \( e_i \): running time error at segment \( i \);
- \( d_{di} \): departure time delay at station \( i \);
- \( T_r \): total reserved time;

**Parameters:**

- \( t_{di} \): actual dwelling time at station \( i \);
- \( t_{ri} \): actual running time on segment \( i \);
- \( u_{\text{max}} \): maximum acceleration;
- \( f_r \): resistance caused by friction;
- \( f_g \): gradient resistance;
- \( f_c \): resistance caused by curvature;
- \( f_d \): interactive impacts among the vehicles;
- \( a, b, c \): Davis resistance parameters;
- \( M \): train mass;
- \( s_{di} \): parking area in station \( i \);
- \( \gamma \): discount factor;

**Decision variables:**

- \( \tilde{t}_{di} \): adjusted dwelling time at station \( i \);
- \( \tilde{t}_{ri} \): adjusted running time at segment \( i \);
- \( u \): accelerating/braking rate.

In general, a subway train operation system basically contains two procedures, as shown in Fig. 1. First, an extensive planning process is rigorously formed into a timetable which is often a long time ahead of real-time operations. The timetable for a certain train can be expressed as \([T_{a_1}, T_{d_1}, \ldots, T_{ai}, T_{di}, \ldots, T_{aN-1}, T_{dN-1}, T_{aN}, T_{dN}]\) and \(T_t = T_{dN} - T_{a_1}\) represents the total trip time. Thus, the planned running time at segment \( i \) and dwelling time at station \( i \) can be written as

\[
\begin{align*}
T_i &= T_{ai+1} - T_{di} \\
W_i &= T_{di} - T_{ai}
\end{align*}
\]

for \(1 \leq i < N\). According to the predetermined timetable, the driver or the ATO controller receives the online information,
which contains the speed, velocity of the train, speed limits and reserved time, and then controls the outputs of train traction/braking systems. From [5] and [9], the subway train operation needs to meet multiple objectives. In detail, it should maintain the safety of the train such that the speed of the train cannot exceed certain limits, reduce the energy consumption, and keep high punctuality and good riding comfort for passengers. The indexes for single-train operations in a certain segment $i$ can be described as follows:

$$Y_i = \begin{cases} 1, & \text{if } v > V_{\text{lim}} \\ 0, & \text{if } v \leq V_{\text{lim}} \end{cases} \quad (2)$$

$$e_{ti} = t_{ri} - T_i \quad (3)$$

$$C_i = \int_0^T \left| \frac{du}{dt} \right| / T \quad (4)$$

$$E_i = \int_0^T \kappa |u(t)| v(t) dt, \quad \text{s.t. } \kappa = \begin{cases} 1, & \text{if } u(t) > 0 \\ 0, & \text{if } u(t) \leq 0. \end{cases} \quad (5)$$

In (2)–(5), we use $Y_i$ to define whether the speed of the train exceeds speed limits. We define running time error $e_{ti}$ as the index of punctuality in (3), which means the error between the actual trip time and planned trip time. For practical experiences in subway operations, it is widely recognized that the train operation method is not applicable if the value of $e_{ri}$ is larger than 10 s. $C_i$ is defined as the indicator of comfort, which implies the total average jerk [40]. The smaller it is, the more comfortable passengers will feel. Besides, the train energy-efficient operation has been given more and more attention as the rising of energy prices and environmental concerns. This is one of our core considerations in designing train control algorithms. The energy consumption between two stations per unit mass is defined as $E_i$ in (5).

However, the real-time train operation is likely to be affected by unexpected disturbances, involving arrival time delay (running time error) and departure time delay which usually make the offline timetable infeasible. In a subway line, running time error is possibly caused by the uncertain train control model, unplanned stop between interstation and even due to some bad weather. Departure time delay is always caused by a long dwelling time because of sudden changes of passenger demands at subway stations. We define running time error and departure time delay at each station as $e_{ti}$ and $d_{di}$, respectively. Consequently, there will be a relatively large time delay $T_{\text{delay}}$ with the predetermined timetable if no decision is made. Thus, considering the unexpected disturbances, the process of real-time train operations in a subway line can be regarded as a decision process given as follows.

- First, when the train is at the initial station, an existing timetable $[T_{a1}, T_{d1}, \ldots, T_{ai}, T_{di}, \ldots, T_{aN}, T_{dN}]$ is made by subway dispatchers;
- If there is no sudden change in the number of transfer passengers at the first station, the dwelling time will be exactly the same as the timetable $W_1$. Otherwise, there will be a departure time delay $d_{d1}$ and the actual dwelling time $t_{d1}$ at the first station is $W_1 + d_{d1}$;
- Before the train departs, a decision needs to be made to determine the planned running times in the following intersections. If there is no big time delay (i.e., $d_{d1} \approx 0$), the driver sets the planned running time the same as the timetable. Otherwise, the driver will change the running times to fit with the timetable as much as possible. In this case, $t_{ri} \neq T_i$;
- After the planned trip time is determined, a corresponding speed profile can be calculated by on-board computers and ATO (drivers) will operate the train to track the speed profile precisely. We define the traction/braking effort with $u \in [-1, u_{\text{max}}]$. The running times on the segment will inevitably be affected by some factors, e.g., different driving behaviors. Considering the running time errors, the actual running time $t_{r1}$ at the first segment is $T_1 + e_{t1}$;
- Then, the train stops at the second station for passengers’ transfer. It receives the real-time information of passenger demands and decides its dwelling times and running times, and we define its reserved time as

$$T_r = T_{dN} - T_{a1} - (W_1 + d_{d1}) - (T_1 + e_{t1}); \quad (6)$$
Consider both the time delay and efficiency, make an appropriate decision to choose the adjusted dwelling time and running time, and select the speed profile accordingly;
• Repeat this process until the train arrives at the destination.

In fact, the quality of real-time train operations may involve several indices in rail traffic management, including the total time delay, energy consumption, riding comfort, and so on. Therefore, considering the trains in the whole subway line with \( N \) stations, (2)–(5) can be reformulated as follows to evaluate the real-time train operation algorithms:

\[
Y = \max(Y_1, \ldots, Y_{N-1})
\]

\[
T_{\text{delay}} = \left[ \sum_{i=1}^{N-1} e_{ti} + \sum_{i=1}^{N} d_{di} \right]
\]

\[
C = \frac{N-1}{N-1} \sum_{i=1}^{N-1} C_i
\]

\[
E_{\text{total}} = \sum_{i=1}^{N-1} E_i.
\]

Therefore, the purpose of real-time train operation algorithms is to develop an online method to meet the following objectives.

• As described above, the current ATO system may not be able to deal with sudden adjusted timetable caused by uncertain disturbances. We aim to realize real-time train operations via a knowledge-based system as well as an online learning approach with real-time data.
• An RTO algorithm only needs online and offline train operation data, and it does not need the precise information of the train model and the predetermined speed trajectory.
• An RTO algorithm should be able to online adjust the current timetable under disturbances, especially uncertain time delay caused by variable passenger demands.
• Once the timetable is adjusted, an RTO algorithm is capable of dynamically changing the control strategies to adapt to the new timetable.
• The algorithm should have high performances with respect to \( Y \), \( T_{\text{delay}} \), \( C \) and \( E_{\text{total}} \).

In the following section, we describe the formulation of the two real-time train operation algorithms. The first algorithm, i.e., the RTO is designed on the basis of a knowledge-based system to meet the multiple objectives. Then, we propose an ITO algorithm based on Q-learning to deal with uncertain disturbances with real-time data, involving the state of the train and passenger demands.

IV. REAL-TIME TRAIN OPERATION ALGORITHMS

A. The Design of RTO

As a train is running on a segment between two adjacent stations, it receives the real-time data of the train, such as speed, position, time and speed limits by multi-sensors (e.g., the radar sensors, acceleration sensors and balises). We use \( \Delta t \) to represent the minimal sample time interval. The current position, speed, and reserved trip time of a train in segment \( j \) are defined as

\[
x_i = [s_i, v_i, T_j - t_i], i = 0, 1, \ldots, m.
\]

Since \( x_0 \) denotes the initial state of the train and \( x_m \) denotes the final state, the following equations should hold:

\[
0 < s_i < S_j
\]

\[
x_0 = [0, 0, T_j]
\]

\[
x_m = [S_j, 0, T_j - t_m]
\]

\[
(0, 0) \xrightarrow{u_0} (s_1, v_1) \xrightarrow{u_1} (s_2, v_2), \ldots, (s_{m-1}, v_{m-1}) \xrightarrow{u_{m-1}} (S_j, 0).
\]

The time interval between each steps is \( \Delta t \). Then, the train control process can be formulated into a discrete-time dynamic system expressed as \( x_{i+1} = F(x_i, u_i) \) for \( \forall i \in [1, m] \), and \( u_i \) denotes the control action and \( F \) is the system function of the train dynamic model.

As described above, the subway train operation needs to consider multi-objectives, and the train control model is very complex with high degree of uncertainty. From real detected data in Beijing Subway, we find that experienced drivers are able to control the train to coast at the appropriate position and distribute the reserved time reasonably such that they can satisfy multiple objectives very well. Hence, a knowledge-based system for the real-time train operation is established by summarizing expert rules from experiences.

The preliminary phase of developing a knowledge-based support system is the selection of the experts. This choice is crucial because the knowledge base is founded on the heuristics and logic of experts. As the selected experts leave a clear imprint on the developed system, we take comprehensive measures to realize knowledge acquisition. For example, we collect an amount of field data in YLBS by different drivers and we choose the data sets with the better driving performance. Then, data mining algorithms, e.g., boosting and ensemble learning are applied. We analyze a survey of onboard passengers from Beijing Subway to study the relationship between riding comfort and the changes in acceleration. In addition, by observing the drivers’ strategies, general knowledge about real-time train operation, along with specific information from the literature [9]–[13], is acquired. Consequently, the knowledge base, which is regarded as a repository of human knowledge, is built with a collection of rules and facts. We list some important expert rules as follows:

1) If the train is starting from the station, then the acceleration should be less than 0.6 m/s\(^2\) to ensure the riding comfort;
2) If the train is running on a segment, then the change rate of acceleration and deceleration should be less than 0.75 m/s\(^3\) and 0.5 m/s\(^3\), respectively, to guarantee the riding comfort;
3) If the speed of the train is lower than speed limit with upward slope, THEN the train should avoid braking as much as possible;
4) If the speed of the train is too low to arrive on time, THEN the train should accelerate to increase the speed;
5) If the train is accelerating, THEN the maximum accelerating value is \( u_{\text{max}} \);
6) If the next speed limit value is lower than the current value, THEN the train should coast or brake in advance to avoid triggering the automatic train protection;
7) If the train approaches a station, THEN the train should keep coasting mode, which is beneficial to comfort of passengers.

Since the real-time train operation is a multi-step control process, any action \( u_i \) may affect not only the immediate performance but also the rewards of the following states. For example, if a train coasts at the starting phase, then it has to accelerate for a much long time to meet the timetable requirements. This means that, it is important to decide to choose the optimal driving strategies by predicting the future states of the train. Reinforcement learning (RL) is a powerful tool in solving a decision-making process with large or continuous state and action spaces. RL is learning how to map situations to actions to optimize a certain reward signal with real-time data from the environment. And it is regarded as a promising approach in realizing learning-based performance optimization of complex dynamic systems, e.g., intelligent transportation system (ITS) [39], [41]. Besides, the expert rules mentioned above can provide constraints for controller’s outputs so that RL with expert rules has better convergence to handle the “curse of dimensionality”.

As described above, the train control process can be formulated as a decision process. (11), (14) define the learning agent’s state and \( u_i \in [-1, u_{\text{max}}] \) defines the finite agent actions. A value function describes the total amount of rewards that an agent can expect to accumulate in the future, starting from the initial state. We consider the average reward in our study:

\[
J_u(x) = \sum_{i=1}^{m-1} \gamma^i r(x_i, u) \bigg/ (m - 1). \tag{16}
\]

The aim of RTO algorithm is to compute the optimal output of train controller \( u^* \) which satisfies

\[
J_{u^*}(x_i) = \min_{u_i} \{ r(x_i, u) + \gamma J^*(x_{i+1}) \} \tag{17}
\]

for all \( x \in X \) and \( u \in U \) such that \( u^* \) has the minimum reward from time \( t_i \). The objective of train operations under a given timetable is to construct an optimal controller output that minimizes the multiple objectives online. It is worth noting that we have derived the expert rules to ensure overspeed protection and running time error. After the description of the knowledge-based system, the specific process of RTO is described in Algorithm 4.1.

**Algorithm 4.1 RTO Algorithm With no Disturbance**

1: Assume that a train departs from station \( j \) and is at state \( x_0 \). The planned running time is \( T_j \) by the current timetable and the distance is \( S_j \);
2: Obtain online data including current train position \( s_i \), velocity \( v_i \) and time \( t_i \);
3: If \( |S_j - s_i| < d_j \), go to 8;
4: Input the online data into the knowledge base and get the feasible solution set \( U \) that satisfies all the expert rules;
5: Improve the policy and the value function by (16)–(18);
6: Output the optimal action \( u^* \) based on

\[
u^* = \min_{u_i \in U} \{ r(x_i, u) + \gamma J^*(x_{i+1}) \} \tag{19}\]

and then update the value function;
7: \( i \leftarrow i + 1 \), go to 2;
8: Brake to park the train at station \( j + 1 \);

**B. The Design of ITO**

For real-world experiences, it is difficult for all the trains to operate precisely matching with the planned timetable. On one hand, the passenger demand varies at different stations and it is difficult to accurately predict the passenger demand variations. If the passenger demand at a station unexpectedly increases, a train has to stay for a longer time for the passengers to transfer. We call this a departure time delay (i.e. the dwelling time delay), expressed as \( d_{di} \) at station \( i \). On the other hand, due to different driving behaviors of drivers, the running times at every segments are uncertain, which causes running time error expressed as \( e_{ti} \). Even though these delays are normally shorter than 20 s, they may result in extra energy usage, accumulative delays and even knock-on delays, especially in a high-density subway line. However, as far as we can see, few of the previous studies of online railway management have considered the uncertainty properties of both departure time delays and running time errors.

The Markov Decision Process (MDP) is a planning framework that allows an agent to reason in the face of uncertainty, optimally trading between actions that gather information and actions that achieve a desired goal. Usually, it can be expressed as a tuple \( \langle X, A, \mathcal{P}, r \rangle \) in which \( X \) and \( A \) are sets of states and actions, \( \mathcal{P} : X \times A \times X \rightarrow [0,1] \) is the state transition probability function, and \( r : X \times A \times X \rightarrow \mathbb{R} \) is the reward function that specifies the immediate reward for each state-action pair.

For a real-time railway traffic management problem, appropriate decisions have to be made to alter both the timetable and the driving strategies in case of disturbances. In the following, we formulate the online timetable adjusting problem into a MDP model with non-deterministic state transition probabilities under uncertain passenger demands and random running...
time errors, and then, we design an integrated real-time train operation (ITO) algorithm based on $Q$-learning to reduce the potential delays and energy consumption.

- **State and Action** In this MDP model, the driver on the train should decide whether the current timetable is the optimal scheme at each decision epoch. To simplify the problem, we assume that decisions are made only when a train reaches each station. It receives the real-time information of passenger demands and its reserved trip time, and then decides its dwelling times and running times in its following journey. If the states of the train does not satisfy the current timetable, the altered dwelling time is denoted as $t_{dk}$ at station $k$, and the altered running time is $t_{rk}$ at segment $k$. Then, the current action $a_k \in A$ is defined as

$$a_k = \left(\tilde{t}_{dk}, \tilde{t}_{rk}, (\tilde{t}_{dk+1}, \tilde{t}_{rk+1}), \ldots, (\tilde{t}_{dN-1}, \tilde{t}_{rN-1})\right)$$  \hspace{1cm} (20)

for all $1 \leq k < N$. Corresponding to the actions, the states $x_k$ of the train at station $k$ is denoted as $x_k = [k, T_{rk}, Z_k]$ where $T_{rk}$ is the reserved trip time and $Z_k$ represents the passenger demand at station $k$. In practice, the operation company defines three kinds of dwelling time delays with no-delay scenario, small-delay scenario and serious-delay scenario because of passenger demand variation. Thus, we use $Z_k \in \{1, 2, 3\}$ to represent the normal passenger demand, increased passenger demand and voracious demand respectively.

- **Reward function** A reward function defines the goal in an MDP model. It maps each perceived state-action pair to a single value, i.e., a *reward* that indicates the intrinsic desirability of that state. Since we focus on online timetable adjusting problem, it is necessary to consider both the subway company’s benefits and the passengers’ benefits, which can be measured by the energy consumption and total time delay, respectively. After a train arrives at station $k+1$ by action $a_k$, the state transfers from $[k, T_{rk}, Z_k]$ to $[k+1, T_{rk+1}, Z_{k+1}]$. As described in [14], the longer trip time will result in lower energy consumption using the same driving strategy. Thus, we define the energy consumption in segment $k$ as $E_k(t_{rk})$ that can be calculate by (5), where $\bar{t}_{rk} = T_{rk+1} - T_{rk} - \tilde{t}_{dk}$. The reward function that reflects the trade-off between energy usage and time delay can be expressed as

$$r_{xx+1}(x_k, a_k) = \sqrt{c_1 D_k(T_{rk}, T_{rk+1})^2 + c_2 E_k(t_{rk})^2}$$  \hspace{1cm} (21)

in which $\bar{D}_k$ and $\bar{E}_k$ represent the normalized values, denoted by

$$\begin{align*}
\bar{D}_k &= (D_{\max} - D_k)/(D_{\max} - D_{\min}) \\
\bar{E}_k &= (E_{\max} - E_k)/(E_{\max} - E_{\min})
\end{align*}$$  \hspace{1cm} (22)

where $E_{\max}$ and $E_{\min}$ are the maximum and minimum values of $E_k$, and $D_{\max}$ and $D_{\min}$ are the maximum and minimum values of $D_k$, i.e.

$$D_k(T_{rk}, T_{rk+1}) = |(T_{ak+1} - T_{ak}) - (T_{rk+1} - T_{rk})|$$  \hspace{1cm} (23)

which means the deviation between the planned timetable and actual running time. In addition, $c_1$ and $c_2$ denote the weight factors of the two objectives, which are the time delay and energy consumption.

- **State transition probability** In this paper, we consider the randomness of subway train operation derived by two factors, i.e., uncertain departure time delays and running time errors. Given any state and action, $x \in X$ and $a \in A$, the probability of each possible next state (i.e., $x'$) is denoted by

$$P_{xx'} = P(x_{k+1} = x' | x_k = x, a_k = a)$$

$$= P((T_{rk} - t_{dk})|x_k, \tilde{t}_{dk}) \times P(x_{k+1} = a)$$

$$= P((T_{rk} - t_{dk}/T_{rk}, \tilde{t}_{dk}) \times P(T_{rk+1}|t_{dk}, \tilde{t}_{rk}),$$  \hspace{1cm} (24)

where

$$T_{rk+1} = T_{rk} - (\tilde{t}_{dk} + \tilde{t}_{rk} + d_{dk} + e_{dk}), 1 \leq k < N.$$  \hspace{1cm} (25)

In addition, it is desirable for the railway managers to impose constraints on real-time train operations. First, the stochastic departure time delay is caused by sudden increase of the number of passengers at busy stations. We use $d_{1d}, d_{2d}$ and $d_{3d}$ to express no-delay scenario, small-delay scenario and serious-delay scenario at each station, and $p_{k1}^d, p_{k2}^d, p_{k3}^d$ to denote the probabilities for $d_{1d}, d_{2d}$ and $d_{3d}$. Note that the probabilities $p_{k1}^d, p_{k2}^d, p_{k3}^d$ are non-deterministic values that may change over time due to some factors, such as pedestrian dynamics. We use $R(x_k, a_k)$ to represent the times that the train is at state $x_k$ and takes action $a_k$. Then, the updating of $p_{k}^d$ is defined as

$$p_{k}^d \leftarrow \frac{p_{k}^d R(x_k, a_k) + 1}{R(x_k, a_k) + 1}, \ \text{if } d_{dk} = d_{dk}^{\epsilon},$$  \hspace{1cm} (26)

for $\epsilon \in \{1, 2, 3\}$. Second, the minimal running time constraints for $\hat{t}_{ri}$ can be calculated by a minimal-time distribution (MTD) algorithm proposed in [9]. Then, we use $A_{x_k} \subseteq A$ to represent the set of possible actions at state $x_k \in X$. According to the constraints given above, we have

$$A_{x_k} = \{a_k \in A | \bar{t}_{ri} \geq \bar{t}_{r1}^\text{min}, \tilde{t}_{di} \in B, k \leq i < N\}$$  \hspace{1cm} (27)

where $B = \{W_i + d_{1d}^3, W_i + d_{2d}^3, W_i + d_{3d}^3\}$ that means the set of possible dwelling time decisions and $T_{r1}^\text{min}$ represents the minimal running time.

After these definitions, we consider a Q-learning approach to solve the MDP problem. $Q$-learning is one of the most important breakthroughs in RL [42], for which the learned action-value function $Q$ directly approximates the optimal action-value function $Q^*$ that is independent of the policy being
followed. A key feature of Q-learning is that it explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment to find an optimal policy. To obtain the Q-function, we use $V_\pi$ to represent the expected value of the cumulative reward received by applying policy $\pi$ from state $x_k$ as

$$V_\pi(x_k) = \mathbb{E}\left[ \sum_{i=0}^{N} \gamma^i r(x_{k+i}, a_{k+i}) \right], 0 < \gamma < 1. \quad (28)$$

Since the value of $Q$ is the reward received immediately upon executing action $a$ from state $x$, plus the value of the following optimal policy, the $Q$-function can be defined as

$$Q(x, a) = \mathbb{E}[r(x, a)] + \gamma \sum_{x' \in X} P_{xx'} a_{x_k} \min_a Q(x', a) - Q(x, a) \quad (30)$$

In addition, we can get $V^*(x) = \min_a Q(x, a)$ from (28) and (30). Therefore, based on the definitions of state, action, reward and state transition probability in (21)–(27), the Q-function updating for timetable adjusting is developed as follows:

$$Q(x, a) \leftarrow Q(x, a) + \alpha \left\{ \mathbb{E}[r(x, a)] + \gamma \sum_{x' \in X} P_{xx'} a_{x_k} \min_a Q(x', a) - Q(x, a) \right\} \quad (31)$$

where $\alpha > 0$ that represents the step-size parameter, and it influences the rate of learning. Finally, based on these definitions and derivations, the process of the ITO algorithm is described in Algorithm 4.2.

Algorithm 4.2 ITO Algorithm Based on Q-Learning

1: Initialize the parameters, input the offline data, and set the train at the first station;
2: Choose the proper dwelling time from $B$ according to the real-time passenger demands;
3: Obtain the reserved time at station $k$. If the time delay with the predetermined timetable is small, go to Step 8;
4: Use (21) and (22) to evaluate the reward at the current state;
5: Compute the $Q$-function by (21)–(27), (29) and (30);
6: Output the altered optimal timetable by $\pi^*(x_k) = \min_a Q^*(x_k, a_k)$;
7: Output the optimal action and update the $Q$-function according to (28)–(31);
8: Use Algorithm 4.1 to operate the train with online data $s$ and $v$ until the train arrives at the next station;
9: $k \leftarrow k + 1$;

V. SIMULATION PLATFORM

To illustrate the effectiveness of the proposed algorithms, we establish a comprehensive simulation platform with the field data in YLBS. The platform consists of three parts, i.e., the input module, the algorithm module and the train model (see Fig. 2). We use the input module to input all the offline
parameters and online data, including the predetermined timetable, speed limits, gradients, running time, speed, position and passenger amounts, etc. The algorithm module is to calculate the optimal driving strategies as well as the real-time adjusted timetable with online data in case of disturbances. The third part, the train model describes the dynamics of train movement by traction, coasting or braking. In YLBS, a type of electric multiple units (EMUs) called DKZ32 are widely used and the control dynamics of DKZ32 can be expressed as

\[ M \ddot{s} = F(s) - f_\tau (s, \dot{s}) - f_g (s) - f_c (s) - f_d (t) \]  

(33)

where \( M \) represents the total weight of the train, \( s, \dot{s} \) and \( \ddot{s} \) represent the position, velocity and acceleration value; \( F(s) \) combines the traction and braking forces; \( f_\tau = a + b \dot{s} + cs^2 \) represents the friction resistance; \( f_g = Mg\sin(\alpha) \) is the gradient resistance and \( \alpha \) is the slope angle; \( f_c = 6.3 M/|r(s) - 55| \) defines the curve resistance and \( r(s) \) is the radius of the curve. Besides, we use \( f_d = \sum_{i=1}^{n-1} (\Delta l_i \sum_{j=i+1}^{n} m_j) \) to describe the varying in-vehicle forces of each vehicle in the EMU, in which \( n \) is the number of vehicles of one EMU, \( m_j \) is the weight of the \( j \)th vehicle for \( 1 \leq j \leq n \) and \( \Delta l_i \) is the spring deformation of the coupler varying [20]. In addition, we use \( \eta, t_p \) and \( t_d \) to represent the system performance gain, time constant and time delay, respectively, which describe the time-delay and non-linearity of the traction/braking instruction transmission [9], [43].

VI. EXAMPLES

In this section, we propose numerical examples to illustrate the effectiveness of the real-time train operation algorithms on a real-life subway line, i.e., Yizhuang Line of Beijing Subway (YLBS) in China. YLBS is an important metro line for both field tests and commercial operations. It has 14 stations starting from Songjiazhuang to Yizhuang with a total length of 23.3 km (Fig. 3). The current cyclic timetable of YLBS in daily operations is shown in Table I and the total trip time \( T_1 \) is 2077 s. The parameters of DKZ32 in YLBS are given as follows. The number of vehicles is \( n = 6 \), and the weights of them are \( m_1 = m_6 = 3.3 \times 10^4 \) kg, \( m_3 = 2.8 \times 10^4 \) kg, \( m_2 = m_4 = m_5 = 3.5 \times 10^4 \) kg; the time constant is \( t_p = 0.4 \), and the time delays for braking and accelerating are 0.8 and 1.0, respectively; \( \Delta l_i = 0.1 \sin(t) \) mm for \( i \in \{1, 3, 6\} \) and \( \Delta l_i = 0.15 \cos(t) \) mm for \( i \in \{2, 4, 5\} \); the maximum accelerating rate is 0.8 m/s²; the Davis parameters \( a, b, c \) are set as 1.244, \( 1.45 \times 10^{-2} \) and \( 1.36 \times 10^{-5} \), respectively. Note that the parameters are solely used for simulations, while the algorithms do not need these information.

Then, we present two different cases to test the algorithms. In Case 1, we consider the deterministic situation to test the performance of RTO. We use the practical train operation data in YLBS, including manual driving data and ATO data. A comparison among the three methods, i.e., manual driving, ATO and RTO, is derived. In Case 2, we consider the subway line with uncertain disturbances for random passenger demands and running time errors, which aims to verify the effectiveness and robustness of the proposed ITO algorithm.

A. Case 1

To test the effectiveness of RTO, we first consider that there is no large delay occurring in the train operation process such that the timetable is unchanged. We choose the practical train operation data without departure time delay from the data sets collected in the YLBS from March 21, 2012 to March 26, 2012. They contain the trains’ position, speed, time, gradients and running time errors, which aims to verify the effectiveness of the real-time train operation algorithms on uncertain passenger demands and running time errors, which aims to verify the effectiveness and robustness of the proposed ITO algorithm.

Fig. 4 shows the velocity/distance (V/D) curves of MD, ATO and RTO. We can see that, all the three driving strategies can keep the velocity of the train being lower than speed limits. Besides, the V/D curve by ATO has many fluctuations when the speed is relatively high. This is caused by the frequently switching of the ATO controller to track the predetermined speed trajectory, whereas the curves of MD and RTO are much smoother, which is beneficial to passengers’ comfort. In addition, the speed of RTO is a little lower than the other two driving strategies and the coasting distance of RTO is longer, which can reduce the energy consumption of train operations.

In this case, we define another performance index, i.e., \( e_t = \sum_{i=1}^{N} |e_{t_i}| \) to express the total running time errors without departure time delays for the whole subway line. The performance values of the three methods are shown in Table II. It is shown that, all the three methods can satisfy the speed limit constraints. Most of the running time errors are within 5 s for the three methods and ATO achieves the best, which is close to the performance of RTO. The running time errors of MD fluctuate badly, which means that manual driving cannot keep punctuality since it is simply based on experience, lacking of rigorous computing. In addition, both RTO and MD outperform ATO in riding comfort. This means that, although driving experiences cannot guarantee punctuality, we can take advantage of
the experiences to enhance the comfort of passengers. Furthermore, ATO consumes the most energy usage among the three methods. The reason may be that ATO has to switch its outputs frequently to precisely track the predetermined trajectory. MD and RTO have better performance in energy consumption and RTO can further reduce about 18% energy usage than MD. This indicates that, experiences by expert drivers are very useful to improve the comfort quality and reduce energy consumption. Besides, by using online learning algorithm, RTO can find the optimal outputs with real-time information from the environment.

### B. Case 2

In this example, we consider the uncertainty properties of dwelling time delays and running time errors to test the effectiveness of ITO algorithm with uncertain disturbances. According to the historical data, we assume that the running time errors at each section are subject to Gaussian distributions. The mean value is zero and the variance is set as 5. In addition, as there are three busy stations including Wenhua yuan, Rongchang and Tongjinan, we assume that the initial delay probabilities for the three stations are $p_{1k} = 0.7$, $p_{2k} = 0.2$ and

<table>
<thead>
<tr>
<th>Station</th>
<th>Songjia zhuang</th>
<th>Xiao cun</th>
<th>Xiao hong men</th>
<th>Jigong</th>
<th>Yizhuang qiao</th>
<th>Wenhua yuan</th>
<th>Wanyuan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival time (s)</td>
<td>0</td>
<td>220</td>
<td>358</td>
<td>545</td>
<td>710</td>
<td>835</td>
<td>979</td>
</tr>
<tr>
<td>Departure time (s)</td>
<td>30</td>
<td>250</td>
<td>388</td>
<td>575</td>
<td>745</td>
<td>865</td>
<td>1009</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Station</th>
<th>Rongjiang</th>
<th>Rongchang</th>
<th>Tongjiaan</th>
<th>Jinhai</th>
<th>Ciqian</th>
<th>Cipu</th>
<th>Yizhuang</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival time (s)</td>
<td>1112</td>
<td>1246</td>
<td>1440</td>
<td>1620</td>
<td>1790</td>
<td>1927</td>
<td>2077</td>
</tr>
<tr>
<td>Departure time (s)</td>
<td>1142</td>
<td>1276</td>
<td>1470</td>
<td>1650</td>
<td>1823</td>
<td>1972</td>
<td>-</td>
</tr>
</tbody>
</table>

![Fig. 4. Velocity and distance curves of MD, ATO and RTO.](image)

**TABLE I**

<table>
<thead>
<tr>
<th>TIMETABLE OF YLBS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Station</strong></td>
</tr>
<tr>
<td>Arrival time (s)</td>
</tr>
<tr>
<td>Departure time (s)</td>
</tr>
</tbody>
</table>

**TABLE II**

<table>
<thead>
<tr>
<th>PERFORMANCE COMPARISON</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
</tr>
<tr>
<td><strong>Driving strategy</strong></td>
</tr>
<tr>
<td>Songjia zhuang - Xiao cun</td>
</tr>
<tr>
<td>Xiao cun - Xiao hong men</td>
</tr>
<tr>
<td>Xiao hong men - Jigong</td>
</tr>
<tr>
<td>Jigong - Yizhuang qiao</td>
</tr>
<tr>
<td>Yizhuang qiao - Wenhua yuan</td>
</tr>
<tr>
<td>Wenhua yuan - Wanyuan</td>
</tr>
<tr>
<td>Wanyuan - Rongjiang</td>
</tr>
<tr>
<td>Rongjiang - Rongchang</td>
</tr>
<tr>
<td>Rongchang - Tongjinan</td>
</tr>
<tr>
<td>Tongjinan - Jinhai</td>
</tr>
<tr>
<td>Jinhai - Ciqian</td>
</tr>
<tr>
<td>Ciqian - Yizhuang</td>
</tr>
<tr>
<td>Overall indices (Y, (\varepsilon_1, \varepsilon_2, \varepsilon_{\text{total}}))</td>
</tr>
</tbody>
</table>
p^3_k = 0.1 with k ∈ {6, 9, 10}. For the other stations, we assume that p^1_k = 0.9, p^2_k = 0.1 and p^3_k = 0. The values of the delays are set as d^1_{2k} = 0 s, d^2_{2k} = 5 s and d^3_{2k} = 10 s, respectively. It is noted that the values of the transition probabilities may change with time and different subway lines. The proposed ITO algorithm can update their values by real-time data in order to capture the dynamic prosperities of passenger demands in a subway line.

For the Q-learning algorithm, γ is the discount rate that determines the present value of future rewards [42]. If γ approaches 1, the objective will be more sensitive to the future rewards, which means that the agent is more farsighted. Since the decision-making process of train rescheduling problem is finite in this paper, we define γ to be 0.98 in Q-learning. In addition, the other parameter of Q-learning, i.e., step-size parameter α, is a small positive fraction. After testing both the permanent and time-varying step-size parameters, we find that the algorithm converges if the parameter is reduced gradually. Thus, the step-size parameter is set as 1/k_s for ITO, where k_s is the iteration steps.

We can see the online learning procedure of ITO in Fig. 5. It is shown that, the total energy consumption is gradually reduced from 2670 to about 2600 after 400 steps. The time delay reaches nearly 9 s at the beginning and it is kept under 1 s after 100 steps. This shows that ITO is able to learn information from an uncertain environment and find a near-optimal policy with real-time data that can reduce both the energy consumption and time delay.

Furthermore, we implement another set of experiments to compare RTO and ITO in the uncertain environment. According to our definition, RTO uses the predetermined timetable all the time while ITO can adjust the timetable online with real-time data. We use the Monte Carlo simulation and run the simulation 300 times for both RTO and ITO with the same parameters. The time delay and energy consumption of the two algorithms are shown in Fig. 6(a) and (b), respectively. It can be observed that the time delays of RTO range from 0 s to 25 s, whereas ITO can keep the time delays within 2 s. In addition, the energy consumption of ITO is a little better than RTO. The average, maximum and minimum of T_{delay} and E_{total} that are defined by (8) and (10), are used to evaluate the two algorithms. In Table III, it is shown that ITO outperforms RTO in all indices. The average time delay of ITO is 0.51 s, which is much better than the average time delay of RTO. It means that, ITO can efficiently overcome the disturbances caused by dwelling time delays and running time errors, and reduce both the energy consumption and time delay via online timetable adjustment.

VII. CONCLUSION

Different from existing studies, this paper proposed two new real-time train operation algorithms via an online learning approach by real-time data in train control process, without using the precise train model and the off-line optimized speed trajectory. First, by summarizing experiences, we built a knowledge-based system which consists of expert rules to ensure the multiple objectives of train operations. Combined with RL, we developed RTO algorithm that uses online data to output the near-optimal train operation strategy. Then, to reduce the time delay and energy consumption of the subway line with uncertain disturbances, we formulated an MDP model with non-deterministic state transition probabilities and designed an ITO algorithm based on Q-learning to online optimize the time delay and energy usage. Finally, we implemented numerical experiments with field data of manual driving and ATO in Beijing Subway, which demonstrates the effectiveness of the proposed approach to handle uncertain factors in a subway line.

With the ubiquitousness of location-based sensors and mobile devices in subway transportation, realizing learning-based performance optimization of ITS in an uncertain environment with real-time data remains to be a challenging task. In this paper, we have combined Q-learning with a knowledge-based system to satisfy the multiple objectives of subway train operations, and handle the uncertainty properties of disturbances with real-time data. Since we only consider some specific departure time delays by passenger demand variations, a tabular representation of Q function is adequate in the implementation of the ITO algorithm. Our future research will focus on the value function approximation [44] in order that we can consider more complex properties of uncertain passenger demands and stochastic running time errors. In addition, the fuzzy approach to handle the uncertain factors can be another topic in our future research.
Fig. 6. Performance comparison with 300 times simulation. (a) Time delay comparison. (b) Energy usage comparison.

<table>
<thead>
<tr>
<th>Performance indices</th>
<th>RTO</th>
<th>ITO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ave $E_{\text{total}}(J/kg)$</td>
<td>2627.4</td>
<td>2616.2</td>
</tr>
<tr>
<td></td>
<td>$T_{\text{delay}}(s)$</td>
<td>5.67</td>
</tr>
<tr>
<td>Max $E_{\text{total}}(J/kg)$</td>
<td>2646.6</td>
<td>2654.1</td>
</tr>
<tr>
<td></td>
<td>$T_{\text{delay}}(s)$</td>
<td>24.52</td>
</tr>
<tr>
<td>Min $E_{\text{total}}(J/kg)$</td>
<td>2506.8</td>
<td>2603.2</td>
</tr>
<tr>
<td></td>
<td>$T_{\text{delay}}(s)$</td>
<td>0.03</td>
</tr>
</tbody>
</table>

References


Jiayin Yin (S’13) received the B.S. degree in 2012 from Beijing Jiaotong University, Beijing, China, where he is currently working toward the Ph.D. degree with the State Key Laboratory of Rail Traffic Control and Safety.

His currently research interests include intelligent control, transportation modeling, and energy-efficient train operation in railway systems.

Dewang Chen (M’06–SM’14) received the Ph.D. degree in control theory from the Chinese Academy of Sciences, Beijing, China, in 2003.

In 2009, he was a Visiting Scholar with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. From 2011 to 2014, he was a Professor with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University. He is currently a Minjiang Chair Professor with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. He is the author of more than 60 papers and two monographs. His current research interests include intelligent control, machine learning, soft computing, and intelligent transportation systems.

Prof. Chen has served on the program committees of more than 15 international conferences and as an Associate Editor for the *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.*