In subway systems, the automatic train operation (ATO) is gradually replacing manual driving for its high punctuality and parking accuracy. But the existing ATO systems have some drawbacks in riding comfort and energy-consumption compared with the manual driving by experienced drivers. To combine the advantages of ATO and manual driving, this paper proposes a Smart Train Operation (STO) approach based on the fusion of expert knowledge and data mining algorithms. First, we summarize the domain expert knowledge rules to ensure the safety and riding comfort. Then, we apply a regression algorithm named as CART (Classification And Regression Tree) and ensemble learning methods (i.e. Bagging and LSBoost) to obtain the valuable information from historical driving data, which are collected in the Beijing subway Yizhuang line. Besides, a heuristic train station parking algorithm (HSA) by using the positioning data storage in balises is proposed to realize precisely parking. By combing the expert knowledge, data mining algorithms and HSA, two comprehensive STO algorithms, i.e., STOB and STOL are developed for subway train operations. The proposed STO algorithms are tested by comparing both ATO and manual driving on a real-world case of the Beijing subway Yizhuang line. The results indicate that the developed STO approach is better than ATO in energy consumption and riding comfort, and it also outperforms manual driving in punctuality and parking accuracy. Finally, the flexibility of STOL and STOB is verified with extensive experiments by considering different kinds of disturbances in real-world applications.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In recent years, urban rail transit has been developed rapidly due to its superiorities of high-speed, punctuality and safety in public transportation systems [1]. Most large cities in China, such as Beijing and Shanghai are building urban metro lines to release the public transport pressure. To improve the efficiency of subway systems, automatic train operation (ATO) has been developed to replace manual driving in many new established subway lines. Since the Victoria line in London Underground network, i.e., the world’s first railway line with an ATO system, was put into operation in 1968, it has become a central issue for researchers to improve the performance of ATO systems. Generally speaking, the existing research in ATO can be divided into three parts, i.e., energy-efficient train operations (ETO) that aim to design energy-saving speed-distance (SD) trajectories, automatic trajectory tracking (ATT) methods to increase punctuality, and train automatic stop control (TASC) to reduce the parking error.

The first part of ATO focuses on energy-efficient train operations (ETO), because traction energy of trains accounts for a large percentage of the total energy-consumption in a subway system [2]–[8]. On the basis of Pontryagin maximum principle, Howlett et al. [2,3] built a discrete train control model and confirmed the fundamental optimality of accelerate-coast-brake strategy for energy-efficient train operations. Considering the variable line gradients and arbitrary speed limits, Khmelnitsky [4] developed a numerical algorithm to obtain the optimal velocity profile. Liu and Golovitcher [5] established a continuous traction force model with constant efficiency to find the optimal control change points on a multi-layer state variable plain. More recently, Albrecht et al. [6] proved that the optimal switching points were uniquely defined for each track segment and the global optimal strategy was unique. In addition to these mathematical solutions, computational intelligent algorithms, such as genetic algorithm, dynamic programming and ant colony optimization have been applied for the calculation of the energy-saving SD trajectories. For example, Chang and Sim [7] applied a genetic algorithm on the ETO problem...
to generate optimal coasting points based on the joint evaluation of punctuality, riding comfort and energy-consumption. Wong et al. [8] illustrated that the genetic algorithm was robust compared with classical calculating methods.

The second topic is the real-time automatic trajectory tracking (ATT) methods, for which ATO controllers are developed to precisely track the predetermined SD trajectory to ensure that the train arrives at the next station on time. Due to the complexity of train dynamic models, many control methods have been proposed to enhance the trajectory tracking accuracy. For example, fuzzy control and predictive fuzzy control, which define some fuzzy rules of train operations, were applied in ATO systems by Yasunobu et al. [9], which was one of the earliest research on ATO systems. Sekine and Nishimura [10] designed a two-degree-of-freedom fuzzy neural network control system, which enabled the trains to operate even under the dynamic variation of train model parameters. Song et al. [12] designed a computationally inexpensive tracking control method where a single-coordinate dynamic model reflecting in-train forces was derived. Gao et al. [13] proposed two robust adaptive control methods considering the actuator saturation and some unknown system parameters. More recently, Li et al. [11] applied a robust cruise control for train operations, and proposed numerical examples that verified the effectiveness of the proposed algorithms to track the SD trajectory precisely.

The third part focuses on the train station parking accuracy, which is also an important index in subway systems, especially for subway stations with platform screen doors (PSD) [15]. Traditional proportional–integral–derivative (PID) based control methods may be difficult to guarantee the parking errors under ± 30 cm, which is a basic requirement that train doors can be opened and passengers can get on and off the train easily. To improve the parking accuracy, Yasunobu et al. [14] designed a fuzzy inference based train stop control algorithm to deal with the uncertain interferences, e.g., resistances, line gradients, etc. Chen et al. [16] proposed three train station parking models, involving a linear model, a generalized regression neural network (GRNN) and an adaptive network based fuzzy inference system (ANFIS), to estimate the train station parking errors. The models were trained through a mass of historical parking data, and the parking model by ANFIS was confirmed to be the most effective to enhance the train parking accuracy. Furthermore, they developed three online learning algorithms [17] to output the optimal train braking rates. Hou [18] applied a terminal iterative learning control (TILC) method into train station stop control, which utilizes the terminal stop position errors in previous braking iterations to update the current control strategy.

By combining ETO, ATT and TASC, an ATO system is able to operate a train automatically with only one supervisor, which greatly reduces manual labors. Besides, since on-board computers of ATO have higher computing accuracy than humans, the punctuality rate of ATO is better than manual driving. Nevertheless, the ATO controllers need to switch frequently in the process of train operations to track the trajectory precisely, which is not conductive to the riding comfort, and reduces the lifetime of ATO controllers. Meanwhile, by analyzing the actual train driving data, we find that, experienced drivers can operate the train to the destination on time by switching the control handle only a few times, which results in lower energy-consumption and better riding comfort.

Generally speaking, the majority of existing ATO systems achieve the train operation goals by focusing on the energy-efficient SD trajectory calculation, real-time tracking methods and train station parking algorithms, separately. Their research seldom considers the multiple objectives such as riding comfort, punctuality, parking accuracy and energy consumption. Moreover, the domain expert knowledge of experienced locomotive drivers play a significant role in designing train control algorithms, which could enhance the service quality and reduce the energy consumption [17,19]. Aiming to improve the overall performance of subway train operations, in this research, we are particularly interested in how to develop Smart Train Operation (STO) algorithms via a system approach by the fusion of expert knowledge and data mining techniques. The contributions of this paper are summarized as follows.

(1) Different from the current research in ATO, this study focuses on the combination of expert knowledge and data mining techniques to realize smart train operations. Since the proposed new STO approach is based on the knowledge from experts and the data recorded from in real-world applications, it can be expected to operation a train without the precise information of the train model and the predetermined SD trajectory.

(2) The existing ATO or manual driving in practice is difficult to satisfy the multiple objectives, i.e., punctuality, riding comfort, parking accuracy and energy consumption. Therefore, we develop two comprehensive STO algorithms that are based on the expert knowledge, data mining techniques and a heuristic train station parking algorithm to improve the overall performances of subway train operations.

(3) Based on the field data in Beijing subway, we derive numerical experiments to analyze the effectiveness and flexibility of the proposed STO algorithms with different kinds of disturbances.

The rest of the paper is organized as follows. Section 2 describes the considered problems, as well as the specific performance indices of train operations in real-world applications. In Section 3, we propose the framework design of STO approach, which includes expert knowledge, data mining methods and a heuristic train station parking algorithm (HSA). In addition, two comprehensive STO algorithms, i.e., STO_P and STO_R are developed by combining expert rules, data mining methods and HSA. In Section 4, an experimental platform is introduced. In Section 5, the effectiveness and flexibility of STO_P and STO_R are tested with extensive experiments by considering different kinds of disturbances. We conclude this paper in Section 6.

2. Problem statement

In subway systems, passenger trains are made up of several electric locomotives, which are powered by electricity from overhead lines, i.e., the pantograph. In general, the handles of electric locomotives can be switched continuously, which means that a driver or an ATO controller can output any orders within the maximum accelerating or braking scope [3]. According to the definition in [22], the aim of subway train operations is to change the switches of train controller appropriately, in order to improve the punctuality, energy consumption, parking accuracy and riding comfort on condition that the speed of the train is under speed limits. We first describe the indices in Eqs. (1)–(5) to evaluate the general performances of subway train operations [7] as follows.

- **Running time error.** We use running time error to evaluate the punctuality of subway train operations on a single segment. Running time error is an important indicator for railway train operation which affects the passengers’ transfer and the timetable of the whole line, i.e.,

\[
\varepsilon_t = t_p - t_r. \tag{1}
\]

It expresses the deviation between the desired running time \( t_p \) and the actual running time \( t_r \). In practice, if the value of \( \varepsilon_t \) is under \( \pm 5 \) s, it is widely recognized that the train operation method has good performance in punctuality.

- **Conversion number of operation modes.** It is abbreviated as \( N_c \), which refers to the switching times that the train controller changes its operational mode. Frequent switches may lower passengers’ comfort and shorten the service life of a train controller.

- **Riding comfort of passengers.** The riding comfort of passengers can be regarded as the subjective feelings of train passengers, which
directly relates to the service quality of a subway system. Based on
the definition in [7], we use $C_r$ to measure the riding comfort
of passengers, shown as follows:

$$C_r = \frac{1}{n} \sum_{i=1}^{n-1} \left| a_i - a_{i+1} \right|$$

where $n$ is the total sampling number, $a_i$ is the controller’s output,
and $\Delta t$ is the sampling time interval. Basically, the smaller $C_r$ is,
the more comfortable passengers feel.

• **Energy consumption per unit mass.** Energy consumption is an im-
  portant indicator due to the rising concerns on energy prices and
  environmental problems. For a subway system, nearly half of the
  energy consumption is used to operate the electric locomotives.

$$\text{Energy consumption per unit mass}$$

$$= \frac{w}{m}$$

where $w$ is the weight of the train, $m$ is the train mass and
$v$ is the velocity of the train.

$$= \int s v \cdot dt$$

where $s$ is the speed and $v$ is the velocity of a train,
and $\Delta t$ is the sampling time interval. Basically, the smaller $C_r$ is,
the more comfortable passengers feel.

• **Parking errors** are required to be within $\pm 30$ cm in Beijing sub-
  way, in order that the PDSs can be opened.

$$e_s = s_p - s_t$$

where $s_p$ is the length of the segment between two adjacent sta-
  tions and $s_t$ is the actual train running distance. Note that the
parking errors are required to be within $\pm 30$ cm in Beijing sub-
way, in order that the PDSs can be opened.

According to the definitions above, the subway train operation
problem is a multiobjective problem [22], which can be described as

$$\min f(a) = (e_r, N_c, C_r, E_w, e_s)$$

subject to

$$a \in A \subseteq \mathbb{R}^n$$

$$s_{i+1}, v_{i+1}) = \mathcal{F}(s_i, v_i, a_i), \forall 1 \leq i \leq n - 1$$

$$v_1 = 0$$

$$s_1 = s_p$$

$$v_i \leq s_p, \forall 1 \leq i \leq n - 1$$

where $a \in \mathbb{R}^n$ is the decision making variable, which expresses
the train controller output, $A$ denotes its feasible set, $(s_{i+1}, v_{i+1})$
represents the speed and velocity of a train, $s_p$ is the length of the interval
between two adjacent stations, $v_l$ is the limited speed, and $\mathcal{F}$ is
the system function of train dynamic model [20,21]. The tradeoff among
the multiple objectives may be inconstant for different times of a day.

For example, a subway company will pay more attention to the punctu-
ality of trains in rush hours, in order that passengers are not af-
fected by the potential delays. Otherwise, energy consumption will
be an important goal to reduce the operation costs.

Generally speaking, the majority of existing ATO systems achieve
the train operation problem by focusing on the energy-efficient SD
trajectory calculation, real-time tracking methods and train station
parking algorithms, separately. For example in Beijing subway, the
process of train operations by ATO can be described as follows. Before
a train departs from stations, the managers of the subway companies
first put forward a set of SD trajectories of the train at every segment
according to the timetable [23]. Then, when the train is running on
a segment, the ATO controller calculates the appropriate traction or
braking output in order that the train tracks the corresponding SD
trajectory precisely. Finally, when the train approaches the next sta-
tion, the ATO controller determines the braking rate and parks the
train at the destination. However, due to the complexity of train dy-
namic models, involving time-varying traction and braking prosper-
ties, variable resistances, etc., this multiobjective problem in Eq. (6)
is difficult to be solved in most existing ATO systems by the classi-
cal optimization approaches. Besides, it usually takes a long time for
subway engineers to calculate the optimal trajectory. This makes it in-
feasible when there are unexpected disturbances, and the timetable
needs to be updated in real-time. Thus, the existing ATO systems are
also lack of flexibility with disturbances caused by variable and un-
certain train dynamic models.

In daily operations of a subway system, there is a large amount
of historical running data stored in train control centers. For current
ATO systems, these important data recording the details of train op-
erations have not been used in full. Besides, we have found that the
experienced drivers have better driving performance in riding com-
fort and energy consumption, while ATO achieves better in punctu-
ality and parking error. This motivates us to study a new approach
that uses the domain expert knowledge and historical driving data
to combine the virtues of ATO and manual driving, in order to solve
this multiobjective problem of subway train operations. (illustrated in
Fig. 1)

### 3. STO: a system approach

In this section, the STO approach for solving the multiobjective
problem of subway train operations is described in details. First, the
components of STO are presented, which are followed by a flowchart
of STO that specifies how it is used in real-world applications. Then,
the expert knowledge, data mining algorithms and HSA are pre-
sent. Besides, the procedures of STO algorithms are developed.

#### 3.1. Description of the STO approach

From Fig. 2, it can be seen that the STO approach contains three
major parts, i.e., expert knowledge, data mining and train station
parking algorithm. The expert knowledge is the foundation of STO, which defines the output constrains of train controllers to ensure the riding comfort and reduce the conversion number of operation modes. Then, we apply data mining algorithms to obtain the appropriate train operation strategies by using the historical driving data. Finally, to enhance the parking accuracy, we propose a heuristic train station parking algorithm (HSA) with the positioning information in balises (a kind of positioning equipments installed in subway stations).

For the STO in real-world applications, it needs two main phases, i.e., the model training phase and the real-time train operation phase. In the model training phase, the collected historical data are selected based on some rules to guarantee the quality of the data. Then, we can obtain the data mining models by training and validation with these selected data. In the real-time train operation phase, a train’s onboard sensors can monitor the real-time information of its position, velocity and running time. Then, we use the expert knowledge to determine a feasible set of its controller outputs. If the train approaches the next station, we use the HSA to park the train. Else, we use the data mining models to obtain the appropriate controller output in real-time. The flowchart of the STO is illustrated in Fig. 3.

3.2. Expert knowledge

Knowledge-based technology is being applied to complex problem solving in many application domains [24]. Due to the complexity of train dynamic model, the most existing ATO methods are difficult to control the trains that have good performances in all the multiple objectives. In addition, the literature [17,22] has shown that manual driving by experienced drivers has better performances than ATO in riding comfort and energy consumption, which motives us to summarize the domain expert knowledge to improve the overall performances of train operations.

We take comprehensive measures to acquire the expert knowledge. For example, we first define several criterions for selection of the experts in our study, which include that the expert should have worked in train operation field for many years and is capable of driving the trains with relatively good performance with guarantee of safety. Then, we derive a series of questionnaires for both experienced drivers and passengers. In addition, we install cameras in the locomotive cabs for observations of driving actions. The summarized expert knowledge can be divided into three parts. The first part is the conversion rules of operation modes, which restrict the operation mode switches to improve the riding comfort. In the second part, we define some operation rules in different operation mode. The third part focuses on a particular case in train operations to ensure that the speed of a train does not exceed the speed limits.

3.2.1. Conversion rules of operation modes

In subway systems, train operations generally include three modes, i.e., traction, coasting and braking. In traction mode, the locomotive produces forward tractive force to accelerate the train, while in braking mode, the locomotive produces backward braking force to slow down the train. In coasting mode, the train relies on inertia to run with the resistances due to the gradient, curvature and wind, etc. For the comfort and safety of passengers, the conversions among the three operation modes should be restricted by the rules described in Table 1.

In Table 1, allowable means that the current operation mode can switch to the next operation mode, while forbidden expresses that it is not allowed for the switch of train operation mode.

3.2.2. Operation rules in different stages

Safety is one of the most important considerations in train operations, which requires that the speed of a train should not exceed
3.3. Data mining methods

In this section, we describe the data mining methods and their applications in locomotive operations. Note that the handles of electric locomotives are continuously. Therefore, the locomotive controller’s output $a$ (from $-1$ to 1, positive is traction, negative is braking, zero is coasting) can take any value in a constrained range $[-1, 1]$. We assume that each value $a \in (0, 1]$ of the output in traction control determines a constant positive acceleration, and it is reasonable to assume that each value $a \in [-1, 0)$ of the output in braking control determines a constant negative acceleration. Similarly, the value $a = 0$ is the output in coasting control. This means that, the locomotive operations can be formulated into a regression problem.

3.3.1. Classification and Regression Trees

Classification and Regression Trees (CART) algorithm [25] was proposed by Breiman et al. in 1984, which is regarded as a major milestone in the evolution of artificial intelligence and machine learning [26]. CART indicates that trees may be used not only to classify entities into a discrete number of groups, but also as an alternative approach to regression analysis in which the value of a response (dependent) variable is to be estimated, given the value of each variable in a set of explanatory (independent) variables.

CART algorithm traditionally involves two steps, that we call growing procedure and pruning procedure. We choose two-thirds of the sample as the training sample which are used in growing procedure, and the rest as the test sample which are used in pruning procedure. The growing procedure permits to construct, from the data, a maximal binary tree $T_{\text{max}}$ by recursive partitioning, and then the pruning procedure permits to select, among all the subtrees of $T_{\text{max}}$, a sequence which contains the entire statistical information.

Ensemble learning is one of the research hotspots in machine learning in recent years, and the main achievements involve Bagging [27], Boosting [28], etc. The main idea of ensemble learning is training multiple weak learning systems and combining the results in a certain way, which can significantly improve the generalization ability of the learning system.

The Bagging algorithm is the most simple and the most intuitive ensemble learning method, which is based on regression trees to dig out the relationship. Theoretically, when using the Bagging algorithm, about 36.8% of the samples will not appear in the new set of training samples averagely as we resample each time. The procedure of Bagging regression algorithm can be seen in Algorithm 1.

**Algorithm 1** Bagging regression algorithm.

**Step 1.** Assume the original set of training samples is $D = \{(x_1, \ldots, y_1), (x_2, \ldots, y_2), \ldots, (x_n, \ldots, y_n)\}$.

**Step 2.** Select $n$ training samples from the original set of training samples at random.

**Step 3.** Train the $n$ samples using the given base learning algorithm, then we can get a model.

**Step 4.** Put back the training samples.

**Step 5.** Repeat $k$ times the above steps from Step 1 to Step 3, so that we can get a set of $k$ models. As for the regression problem, we can obtain a final forecasting model as follows:

$$F(x_i) = \frac{1}{K} \sum_{k=1}^{K} f_k(x_i).$$

Boosting was proposed for classification tasks, but it can be used to solve the regression problems after appropriate modification. The earliest Boosting regression algorithm is AdaBoost [28], proposed by Freund and Schapire, is one of the most important ensemble methods due to its theoretical foundation and great simplicity. Then, Drucker [29] improved AdaBoost and he proposed LSBoost, which can be used for regression problems. We describe the procedure of LSBoost regression algorithm in Algorithm 2.

| Table 1 Conversion rules of operation modes.  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COM$^a$</td>
<td>NOM$^b$</td>
<td></td>
</tr>
<tr>
<td>Traction</td>
<td>Coasting</td>
<td>Braking</td>
</tr>
<tr>
<td>Traction</td>
<td>None</td>
<td>Allowable</td>
</tr>
<tr>
<td>Coasting</td>
<td>Allowable</td>
<td>None</td>
</tr>
<tr>
<td>Braking</td>
<td>Forbidden</td>
<td>Allowable</td>
</tr>
</tbody>
</table>

$^a$ COM: current operation mode.  
$^b$ NOM: next operation mode.

![Fig. 4. An example for the descending of limited speed.](image-url)
Algorithm 2 LSBoost regression algorithm.

Step 1. For the original training set \( \{(x_i, y_i), i = 1, \ldots, m\} \), determine the base learning algorithm and the ensemble size.

Step 2. Distribute one equal weight on all training samples \( w_i^{(t)} = 1/m, i = 1, 2, \ldots, m \) and set up a member regression model \( f_i^{(t)} \).

Step 3. Use base regression model \( f_i^{(t)} \) going through all samples of the training set to get the predicted value of every sample. Calculate the sample error, the function in square form defined as:

\[
I_i^{(t)} = \frac{\left| y_i^{(t)} - y_i \right|^2}{\max_j \left| y_i^{(t)} - y_j \right|^2} (i, j = 1, 2, \ldots, m).
\]

Step 4. Calculate the average error of \( f^{(t)} \) in the set of training samples: \( \bar{L}^{(t)} = \sum_{i=1}^{m} L_i^{(t)} \). If \( \bar{L}^{(t)} \geq 0.5 \), then turn to Step 3. Otherwise, go to Step 5.

Step 5. Calculate \( f^{(t+1)} = f^{(t)} + \beta \sum_i \omega_i^{(t+1)} y_i \). If \( t = T \), then go to Step 6. Otherwise, update the weights of the training samples \( \omega_i^{(t+1)} = \omega_i^{(t)}(1 - L_i) \) and \( t = t + 1 \), then turn to Step 2.

Step 6. For each new input sample \( x_i \) and every regression model give a predicted value \( y_i^{(t)} (t = 1, 2, \ldots, T) \). Order the \( T \) predicted values from small to large \( y_1^{(t)} \leq y_2^{(t)} \leq \ldots \leq y_T^{(t)} \), where \( k_1, k_2, \ldots, k_t \in [1, T] \). Then cumulate \( 1/\beta^t \) in the order of the predicted values from small to large until \( y \in [1, T] \), which meets the requirements:

\[
\sum_{t=1}^{T} \log(1/\beta^t) \geq \frac{1}{2} \sum_{t=1}^{T} \log(1/\beta^t)
\]

where \( y_T^{(t)} \) is the final predicted value of the sample \( i \), and in fact, it is the weighted median of \( T \) predicted values.

3.4. Heuristic train station parking algorithm

In the train stopping stage, we consider the precise positioning information provided by balises, which play an important role in train control systems [17]. In a subway platform, the engineers install several balises on the tracks in front of the train parking point. When a train approaches the station and goes through each balise, the train can receive the precise positioning data from the balise. According to the field data in Yizhuang Line, we find that there are 5 balises installed in the stopping stage and the last balise is installed at the train parking point. We use \( S_i \) for \( i \in \{1, 2, 3, 4, 5\} \) to represent the distance from balise \( i \) to the parking point. The positions of the balises can be represented as \( S_1 = 102 \text{m}, S_2 = 58 \text{m}, S_3 = 13 \text{m}, S_4 = 6 \text{m} \) and \( S_5 = 0 \text{m} \). We define \( S_0 = S_1 \), which represents the distance of train parking stage. Then, the calculating formula of train controller's output between two balises is given as follows.

\[
\begin{align*}
\begin{cases}
\alpha_i = -\frac{v^2}{2S_1} (i = 1), \\
\alpha_i = -\frac{v^2}{2S_i} - \eta \alpha_i^{a+1} (i = 2, 3, 4) \\
\alpha_5 = -1
\end{cases}
\end{align*}
\]

\[
\alpha_i^a = -\frac{v_i^2 - v_(i-1)^2}{2S_i - S_(i-1)} (i = 1, 2, 3)
\]

\[
\alpha_i^v = \alpha_i^a - a_i (i = 1, 2, 3)
\]

where \( v_i \) is the actual speed when passing each balise, \( a_i \) is theoretical output of controller, \( \alpha_i^a \) is actual output of controller, \( \alpha_i^v \) is derivation of controller's output and \( \eta \) is the learning rate.

3.5. Procedure of STO algorithms

Based on the aforementioned descriptions of expert knowledge, data mining and HSA, the STO algorithms will be developed in what follows. The procedure given in Algorithm 3 illustrates the overall

Algorithm 3 The STO algorithms.

Step 1. Input the historical train driving data, and select the data with good performances.

Step 2. Express the selected data as

\[
\begin{pmatrix}
\begin{array}{cccc}
\gamma_1 & \gamma_2 & \cdots & \gamma_n \\
\delta_1 & \delta_2 & \cdots & \delta_n \\
\end{array}
\end{pmatrix}
\]

\[
\begin{pmatrix}
\begin{array}{cccc}
x_1^1 & x_1^2 & \cdots & x_1^7 \\
x_2^1 & x_2^2 & \cdots & x_2^7 \\
\cdots & \cdots & \cdots & \cdots \\
x_n^1 & x_n^2 & \cdots & x_n^7 \\
\end{array}
\end{pmatrix}
\]

\[D = \begin{pmatrix}
\begin{array}{cccc}
\gamma_1 & \gamma_2 & \cdots & \gamma_n \\
\delta_1 & \delta_2 & \cdots & \delta_n \\
\end{array}
\end{pmatrix}
\begin{pmatrix}
\begin{array}{cccc}
x_1^1 & x_1^2 & \cdots & x_1^7 \\
x_2^1 & x_2^2 & \cdots & x_2^7 \\
\cdots & \cdots & \cdots & \cdots \\
x_n^1 & x_n^2 & \cdots & x_n^7 \\
\end{array}
\end{pmatrix}
\]

in which \( n \) represents the groups of data and \( x_1^1, x_2^1, \ldots, x_n^7 \) represents the limited speed, gradient, train speed, remaining time, remaining distance, changing value of the next limited speed, and the remaining distance of next limited speed.

Step 3. Train the \( n \) samples using the given base learning algorithm by Algorithm 1 or Algorithm 2, then we can get a data mining model, B-CART or L-CART by Bagging or LSBoosting, respectively.

Step 4. Input the online data of train operations, which include the speed \( v \), position \( s \), speed limit, gradient and reserved time.

Step 5. If \( s > S_n \), turn to Step 9, else go to Step 6.

Step 6. Update the feasible set \( A \), i.e., a set of values that the controller can take according to the expert knowledge.

Step 7. Use B-CART or L-CART model to obtain the optimal controller outputs from \( A \).

Step 8. Update the real-time information of the running train, and then go to Step 4.

Step 9. Use HSA in Section 3.4 to park the train at the station.

flowcharts in Fig. 3. In Algorithm 3, Steps 1–3 describe the model training phase, and Steps 4–9 describe the real-time train operation phase. Thus, by the fusion of expert knowledge and data mining algorithms, we obtain two STO algorithms, i.e., STo and STO, which are based on Bagging (Algorithm 1) and LSBoost (Algorithm 2), respectively.

4. Description of the experimental platform

To evaluate the effectiveness and flexibility of the proposed STO algorithms, we establish an experimental platform based on the real-world data from Yizhuang Line, Beijing subway (YLS). It can simulate the running of locomotives on a subway line with given driving strategies. First, we present the train dynamic model in the experimental platform, which is described as

\[
M \frac{dv}{dt} = Ma - f(s, v).
\]

\[
f(s, v) = Mg \sin(\alpha) + \lambda_1 v^2 + \lambda_2 v + \lambda_3 + 6.3M [r(s) - 55]
\]

where \( M \) is the weight of a train and \( f(s, v) \) represents the resistances. The resistances are made up of gradient resistance \( Mg \sin(\alpha) \) where \( \alpha \) is the slope angle, track resistance \( \lambda_1 v^2 + \lambda_2 v + \lambda_3 \) where \( \lambda_1, \lambda_2 \) and \( \lambda_3 \) are Davis parameters, and curve resistance \( 6.3M [r(s) - 55] \) where \( r(s) \) is the radius of the curve [30]. In addition, there are nonlinearity and time delay in a train accelerating/braking system [22]. We use \( T_c, T_d, B_c \) and \( B_d \) to represent the traction time constant,
5.1. Experimental data

traction time delay, braking time constant and braking time delay, respectively. Note that the type of locomotive used in Yizhuang line is traction time delay, braking time constant and braking time delay, respectively. Note that the type of locomotive used in Yizhuang line is DKZ32 and the parameters of DKZ32 are showed in Table 2.

5. Numerical results and comparisons

In this section, we derive numerical experiments based on the real-world train operation data in YLBS. We first describe the data collection process, as well as the expert rules that are used for data selection. Based on the historical train operation data, the data mining models are trained and analyzed. Then, we develop a real-world case of YLBS to test the effectiveness of the proposed STO algorithms by comparing with both ATO and manual driving. Finally, the flexibility of STO algorithms is verified by dynamically adjusting the parameters of locomotive models.

5.1. Experimental data

The data used in this study are collected in Yizhuang Line of Beijing subway (YLBS) (illustrated in Fig. 5), which connects the center of Beijing and Yizhuang Economic Development Zone, and the length of this line is 23.23 km. The field data are collected from the on-board computers of locomotives (Fig. 6), which record the running data of a train by speed sensors. The data contains the detailed information of the train’s position, velocity, running time, acceleration and braking rates, speed limits and line gradients. In addition, the sample interval is 0.2 s, which indicates that the data are recorded every 0.2 s.

We choose one segment between Rongjing East Street and Wanyuan Street as an example. On this segment, we collected 11,235 groups of data sets throughout two days. We calculate the performance indices for the collected data sets by Eqs. (1)-(5), and the results are illustrated in Table 3. Due to the psychological or physical conditions of drivers, such as fatigue and pressure, we can find that some of these collected data sets are with relatively low performance. Therefore, we adopt several selection rules on the basis experiences to make sure that the developed STO approach can be effective.

- Rule 1: Running time error is within ±5 s.
- Rule 2: Parking error is within ±30 cm.
- Rule 3: Conversion number of operation mode is within 10 times.
- Rule 4: Riding comfort is within 0.08 m/s³.
- Rule 5: Energy consumption is less than 210 J/kg.

Then, we select 7312 groups of data sets that satisfy the above rules from the original data sets. In the following, we use the 7312 groups of data sets to train the data mining models.

5.2. Data mining models

Now we use the training algorithms described in Algorithm 1 and Algorithm 2 to train the selected 7312 groups of data sets. Note that we use CART algorithm as the weak learning machine of Bagging and LBoost in Algorithm 1 and Algorithm 2, which are called B-CART and L-CART, respectively. In addition, we also use an algorithm that is based on single CART for comparison. The iteration step is set 100 in the training process and the absolute errors of the three data mining algorithms are shown in Fig. 7.

In Fig. 7, compared with single CART, both B-CART and L-CART get better performance after about 10 iterations, and L-CART achieves the best performance. The results are coincident with the theoretical analysis. Besides, it can be seen that the mean absolute errors of B-CART and L-CART are almost constant after about 50 iteration steps. So the upper bound of iteration times is set as 50 in our following experiments.

Furthermore, we illustrate the regression models of B-CART and L-CART in Figs. A.1 and A.2 in Appendix A. In Figs. A.1 and A.2, $x_1$, $x_2$, …, $x_7$ represent the speed limit, train speed, gradient, remaining distance, remaining time, value of next speed limit interval and remaining distance of next speed limit interval, respectively. It can be seen that the regression model of STO_b is larger than the regression model of STO_g, and the numbers of leaf nodes in the two models are 19 and 30, respectively. We define that a root node to each leaf

| Table 2 |
|-----------------|-----------|
| Parameters | Value |
| Time constant (Braking) $T_s$ | 0.4 |
| Time delay (Braking) $S_t$ | 0.8 |
| Time constant (Accelerating) $T_a$ | 0.4 |
| Time delay (Accelerating) $S_a$ | 1 |
| $M$ (kg) | $2 \times 10^3$ |
| $\lambda_1$, $\lambda_2$, $\lambda_3$ | $(1.36 \times 10^{-4}, 1.45 \times 10^{-2}, 1.244)$ |

| Table 3 |
|-----------------|-----------|
| Indices | Variation range |
| Running time error (s) | [−5.2, 7.8] |
| Conversion number (time) | [4, 16] |
| Riding comfort (m/s³) | [0.047, 0.096] |
| Energy consumption (J/kg) | [197.32, 216.63] |
| Parking error (cm) | [−11 242, 4.74] |
5.3. Effectiveness of STO algorithms

In this experiment, we use the trained data mining models to test the effectiveness of the STO algorithms. Taking the interstation between Rongjing East Street and Wuanyuan Street as an illustrative example. The distance between the two stations is 1280 m, and the desired trip time is 101 s. Fig. 8 shows the gradients of this segment. In addition, we also choose two sets of field data in YLBS that record the operation curves by ATO and manual driving for comparison. The ATO system in YLBS uses an adaptive PID controller with some output constraints to track a predetermined SD trajectory.

The comparisons of STOβ, STOL, ATO and manual driving are shown in Figs. 9 and 10. It can be seen from Fig. 9 that, the SD curves of STOβ, STOL and manual driving are very similar, while the SD curve of ATO has some fluctuations. This indicates that the STO algorithms can be effective to learn from manual driving data to guarantee the steady of train speed. The controller outputs of ATO, manual driving, STOβ and STOL are shown in Fig. 10. It can be seen that the ATO controller switches very frequently, and the outputs of the manual driving, STOβ and STOL are smooth and steady, which means that the STO approaches may improve riding comfort of passengers by experiences and historical driving data.

The performances of the four driving methods, calculated by Eqs. (1)–(5), are shown in Table 4. It can be seen that: (1) $N_c$, $C_r$ and $E_u$ of STO algorithms are much better than ATO, which means that the STO algorithms can save energy and enhance the riding comfort by changing the output of train controller only a few times; (2) the running time error $e_t$ of ATO is better than manual driving and STO algorithms, which indicates that ATO controller has good punctuality by rigorous computation; (3) nevertheless, the overall performances of STO algorithms outperform both ATO and MD. Besides, STOβ has the best performance on $e_t$, $N_c$, $C_r$ and $E_u$, and its running time error is less than 3 s, which also satisfies the requirements of subway operations. In addition, the results also illustrate that, ATO focuses on the objective of punctuality and MD is better in riding comfort. Meanwhile, STOβ and STOL can obtain better trade-off solutions, since they have larger objective weights for punctuality and riding comfort, respectively.

![Fig. 7. Mean absolute errors with iteration times.](image7)

![Fig. 8. Gradient between Rongjing East Street and Wuanyuan Street.](image8)

![Fig. 9. Speed-distance curve comparison.](image9)

![Fig. 10. Controller output comparison.](image10)

Table 4

<table>
<thead>
<tr>
<th>Index</th>
<th>$e_t$ (s)</th>
<th>$e_t$ (cm)</th>
<th>$N_c$ (time)</th>
<th>$C_r$ (m/s²)</th>
<th>$E_u$ (J/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATO</td>
<td>1.21</td>
<td>3.00</td>
<td>14.00</td>
<td>0.42</td>
<td>236.7</td>
</tr>
<tr>
<td>MD*</td>
<td>2.90</td>
<td>9.51</td>
<td>8.00</td>
<td>0.12</td>
<td>205.3</td>
</tr>
<tr>
<td>STOβ</td>
<td>1.46</td>
<td>1.81</td>
<td>7.00</td>
<td>0.04</td>
<td>206.5</td>
</tr>
<tr>
<td>STOL</td>
<td>2.70</td>
<td>1.45</td>
<td>3.00</td>
<td>0.04</td>
<td>197.6</td>
</tr>
</tbody>
</table>

* MD: manual driving.
5.4. Flexibility analysis

Due to the complexity of train dynamic models, the parameters of locomotives, which including \( T_c, T_d, B_c \) and \( B_d \), may be uncertain and dynamic because of the variation of external environment. For example, in an underground tunnel, there is much moisture, which will increase the time constant and time delay of a train. Therefore, it is important for the STO algorithms to be flexible with variable parameters of train dynamic models.

To evaluate the flexibility of STO algorithms, we define four statistical indices as follows. 1) Mean, mean value of performance index; 2) Min, minimum of performance index; 3) Max, maximum of performance index; 4) RMSE, root mean square error of performance index, which are expressed in Eqs. (18)–(21).

\[
\text{Mean} = \bar{p} = \frac{1}{m} \sum_{i=1}^{m} p_i
\]  
\[
\text{Min} = \min(p_1, p_2, p_3, \ldots, p_m)
\]  
\[
\text{Max} = \max(p_1, p_2, p_3, \ldots, p_m)
\]

\[
\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{m} (p_i - \bar{p})^2}{m}} \tag{21}
\]

where, \( m \) is the number of simulations, \( p_i \) is the \( i \)-th value in the experiments of the four performance indices. In the following, we first analyze the impact of single parameter variations on STO algorithms. Then, we change multiple parameters of train dynamic model simultaneously to test the robustness of STO algorithms.

5.4.1. Single parameter variation

In this section, we analyze the impact of individual parameter on STO algorithms. The fluctuation of the parameters is set as 20% and
Fig. 13. Relationship between $T_d$, $T_c$ and $e_t$, $e_s$ of STO$_b$.

Fig. 14. Relationship between $T_d$, $T_c$ and $e_t$, $e_s$ of STO$_c$.

Fig. 15. Relationship between $B_d$, $B_c$ and $e_t$, $e_s$ of STO$_c$. 
between the two samples. If correlation. If $r = 0$, it indicates that there is no correlation between the two samples.

In Figs. 11 and 12, with the increase of $T_d$ and $T_c$, the parking error $e_t$ fluctuates in a certain range and the running time error $e_t$ increases slightly, while $e_t$ increases greatly with the increase of $B_d$ and $B_c$. This indicates that, there are some relationship between $e_t$ and the parameters of train dynamic model. Therefore, we define a correlation coefficient $r$ in the following equation to analyze the correlation of them.

$$ r = \frac{\sum_{i=1}^{N} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N} (X_i - \bar{X})^2 \sqrt{\sum_{i=1}^{N} (Y_i - \bar{Y})^2}}} \tag{22} $$

where $X$ is the parameter, $Y$ is $e_t$ or $e_s$, and $r \in [-1, 1]$. If $r = 0$, it indicates that there is no correlation between the two samples. If $|r| = 1$, it indicates that the two samples are completely linear correlation. If $0 < |r| < 1$, it indicates there is a nonlinear correlation between the two samples. If $r > 0$, it indicates that the two samples have a positive correlation (PC). Otherwise $r < 0$, it indicates that the two samples have a negative correlation (NC). The greater strength of the association results in the larger absolute value of correlation coefficient. Here we assume that, if $|r| > 0.8$, it indicates that the two samples are correlation(CO); if $|r| < 0.3$, it indicates that the two samples are irrelevant(IR). The correlation of parameters for two STO algorithms is shown in Table 5.

In Table 5, the two STO models are consistent and the four parameters have a positive correlation with $e_t$. Moreover, $B_d$ and $B_c$ have a positive correlation with $e_s$. This is accord with the practical situation, since variations of parameters in the train braking system (i.e. $B_d$ and $B_c$) will affect the parking error.

5.4.2. Robustness tests by changing multiple parameters

In this part, we try to test the robustness of STO algorithms by changing multiple parameters simultaneously. First, we change two parameters at the same time. The fluctuation of the parameters and the increment for each parameter are set as 20% and 1%, respectively. We run the experiments by $41 \times 41$ times, and we also take $e_t$ and $e_s$ as examples. The results are shown in Figs. 13–16.

In Figs. 13 and 14, the running time error $e_t$ of STOB increases accordingly, and the parking error $e_t$ of STOB and STOL fluctuates slightly. In addition, the values of $e_t$ and $e_s$ for two STO algorithms are all within the allowed range. For example, the running time errors are kept within 5 s with the variations of parameters. This reflects the STO algorithms’ robustness with variation of traction parameters. Similarly, Figs. 15 and 16 indicate the robustness of STO algorithms with variable parameters of train braking system.

Then, we assume all variances have 20 percent changes (80–120%) for the four parameters of train traction and braking system. We set increment for each parameter as 10%, which means that the values of parameters are as follow: $T_d \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$; $T_c \in \{0.32, 0.36, 0.4, 0.44, 0.48\}$; $B_d \in \{0.64, 0.72, 0.8, 0.88, 0.96\}$; $B_c \in \{0.32, 0.36, 0.4, 0.44, 0.48\}$. Then, we run the experiments for 625 times with STOB and STOL. The numerical results are shown in Table 6.

![Fig. 16. Relationship between $B_d, B_c$ and $e_t, e_s$ of STO.](image)

![Table 6. The statistical results of robustness.](table)

<table>
<thead>
<tr>
<th></th>
<th>STOB</th>
<th>STOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>$e_t(s)$</td>
<td>1.36</td>
<td>4.2</td>
</tr>
<tr>
<td>$e_s(cm)$</td>
<td>3.881</td>
<td>9.527</td>
</tr>
<tr>
<td>$N_e$</td>
<td>6.385</td>
<td>16</td>
</tr>
<tr>
<td>$C_e(m/s^3)$</td>
<td>0.202</td>
<td>0.303</td>
</tr>
<tr>
<td>$E_e(J)$</td>
<td>206.56</td>
<td>210.55</td>
</tr>
</tbody>
</table>
are not affected by varying parameters. For example, the running time errors are less than 5 s for all the instances. The above experiments conclude that, by the fusion of expert knowledge and data mining algorithms, the developed STO algorithms can meet the requirements of train operations, and they achieve better than ATO and manual driving. Furthermore, the robustness tests indicate the good performances with variable parameters of the STO algorithms.

6. Conclusion

To combine the advantages of ATO and manual driving, we proposed a system approach to achieve smart train operations via integrating expert knowledge, data mining methods and a train parking algorithm. Field data in manual driving of excellent drivers were collected from Beijing subway Yizhuang line. Two STO algorithms i.e. STOB and STOL were developed by the fusion of expert knowledge and data mining methods. Finally, the STO algorithms were tested through an experimental platform with real-world data sets to test the effectiveness and flexibility of the developed algorithms.

From the results, the two STO algorithms have achieved good performances by comparing with ATO and manual driving, especially in riding comfort and energy consumption. In addition, STOL performs the best in the overall indices. The flexibility of the STO approach is verified with a large number of experiments by considering four disturbances of train traction and braking systems.

Even though the STO algorithms are shown to be effective in this paper, there are still some issues to be addressed in our future research. For example, although the STO algorithms are adopted to some general lines in real-world cases, the lines with steeper slope or complex limited speed are needed to be studied. Moreover, more advanced machine learning algorithms (deep learning, transfer learning [31]) are worth further research for real-world scenarios.

Acknowledgments

This work has been partially funded by the Beijing Municipal Natural Science Foundation under Grant 4142044, by the Start Funding for Minjiang Chair Professor by Fujian Province under Grant 510146, and by the State Key Laboratory of Rail Traffic Control and Safety under Grant RCS2014ZZ02. The authors thank the anonymous referees for constructive comments which greatly improve the quality of this paper.

Appendix A. Illustrations of regression models of STOB and STOL

Fig. A.1. Regression model of STOB.
Fig. A.2. Regression model of STOL.

References