Hindawi
Journal of Advanced Transportation
Volume 2023, Article ID 1217352, 12 pages
https://doi.org/10.1155/2023/1217352

Research Article

Study on Energy-Saving Train Trajectory Optimization Based on Coasting Control in Metro Lines

Bo Jin,[1,2] **Song Yang**,[1] **Qingyuan Wang**,[3] **and Xiaoyun Feng**[3]

[1]*Zhejiang Scientific Research Institute of Transport, Hangzhou 310023, China*
[2]*Institute of Intelligent Transportation Systems, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China*
[3]*School of Electrical Engineering, Southwest Jiaotong University, Chengdu 611756, China*

Correspondence should be addressed to Bo Jin; kimbojin@163.com

Received 4 November 2022; Revised 3 January 2023; Accepted 17 January 2023; Published 22 March 2023

Academic Editor: Luca Pugi

With increasing energy consumption in urban rail transit systems, researchers have paid significant attention to energy-saving train control. In this paper, we propose an effective train trajectory optimization method to reduce the energy consumption based on coasting control, in which coasting control regimes are added to balance running time and energy consumption. For better determining the starting points of coasting control regimes, the whole train running process is divided into several subintervals. Then, aiming to achieve energy efficiency, coasting regimes are added to the subintervals with high energy-saving effects, in which more energy consumption can be reduced with the same running time addition. Based on this, a coasting control method is proposed to generate energy-saving trajectories considering train dynamics, safety, and punctuality. In addition, the proposed method can solve the multisection energy-saving train trajectory optimization problem to obtain optimal running time schemes and related trajectories. Finally, numerical examples based on one of the Beijing metro lines are implemented to verify the effectiveness of the proposed method. The results show that, for the single-section train control problem, the proposed coasting control algorithm can achieve significant energy-saving effects compared to the practical trajectory and calculate energy-saving trajectory in shorter computation times compared to the dynamic programming method. Meanwhile, for the multisection train control problem, energy consumption can be further reduced by optimizing trajectories and running times integratedly.

1. Introduction

Urban rail transit (URT) systems are developing rapidly in recent years to meet the increasing passenger demands. Meanwhile, URT systems consume a huge amount of energy, especially in big cities (e.g., Beijing, New York, and Tokyo). With rising energy prices and environmental issues, energy cost is becoming a grand challenge. Therefore, energy-saving strategies are being implemented to reduce energy consumption. These strategies mainly include [1] applying the energy-efficient rolling stock; demand-driven train timetabling aiming to reduce the number of train services; energy-efficient train timetabling; and energy-efficient train control. In this paper, we focus on the energy-efficient train control. More details about other strategies can be found in works [1, 2].

Energy-efficient train control mainly focuses on reducing energy consumption by optimizing train trajectory (or called speed profile and driving strategy). In recent years, many works have been devoted to design algorithms for generating energy-saving train trajectories while satisfying operational constraints. The first research on the optimal train control problem was carried out in 1968, in which Pontryagain's maximum principle (PMP) was used to solve the problem for level tracks [3]. Based on the PMP, the optimal train control regimes (i.e., maximum acceleration (MA), cruising (CR), coasting (CO), and maximum braking (MB)) for the energy-efficient operation were proposed. In addition, many researchers have applied the PMP to solve the optimal train control problem considering varying speed restrictions and gradients [4–7]. Especially, the scheduling

and control group at the University of South Australia presented a systematic review of the optimal train control theory from the viewpoint of PMP [8, 9]. Methods in studies [3–9] belong to the indirect methods, in which the optimal solutions are obtained with complicated computational processes and large computation times.

With the application of automatic train operation (ATO) systems in railway systems, especially in URT systems, the recommended trajectory for the next running process should be determined before train departure. Thus, direct methods are applied to calculate optimal train trajectories with shorter computation times. In direct methods, the control actions (i.e., traction/braking force or acceleration) and/or state variables (i.e., position, speed, and time) are discretized to transform the optimal train control problems into mathematical programming problems [10]. By splitting the train running process to build a discrete-position model and linearization with piecewise approximation, the original optimal train control problem was rebuilt into a mixed integer linear programming model, which can be solved effectively by existing solvers [11, 12]. Dynamic programming (DP) algorithm has been widely applied in the optimal train control problem [13–17]. It was necessary to transform the optimal problem into multiple decision processes in the DP algorithm, which can be realized by state space discretization [13]. Specially, for the energy-efficient train control problem, the cost of state transition was set as energy consumption and decision actions were set as traction and braking forces. By backwards calculating the optimal policies of each state space point and forwards searching, a set of optimal control actions and the optimal trajectory with minimum energy consumption can be obtained. In study [14], the performance of the DP algorithm, genetic algorithm, and ant colony optimization algorithm was contrasted and compared. It was found that the DP algorithm can obtain the best solution with more computational resources compared to the other two algorithms. The pseudospectral method was also employed in the optimal train control problem, in which the continuous-time optimal control problem was transcribed into a discrete nonlinear programming problem [18, 19].

On the other hand, some works focus on determining the optimal conversion points of control regimes to formulate the energy-efficient train trajectories. Train trajectory optimization based on coasting control is a popular methodology to enhance the energy efficiency, in which energy consumption is reduced by adjusting coasting regimes. Generally, in coasting control, the starting points of coasting regimes are determined by genetic algorithms [20–23]. A genetic algorithm was proposed to search for the points of coasting regimes where the number of coasting regimes was predetermined [20]. To deal with complex operation situations, a hierarchical genetic algorithm was introduced to integrate the determination of the number of coasting regimes and points of coasting regimes [21]. The simulation results showed that coasting control was an economical approach to balance running time and energy consumption, which can achieve a good performance in energy-saving. However, genetic algorithms may fail to converge onto a good solution and cause a long computation time. In study [23], coasting control was applied to calculate energy-saving trajectories, considering the utilization of regenerative braking energy.

Different from the existing coasting control methods [20–23], this paper proposes a novel searching method to determine the points of coasting regimes. First, we formulate a distance-based model to describe the train dynamics. Considering the operational constraints and the energy-saving objective, an optimal train control model is built. Then, flay-out trajectory and subinterval are introduced. The former clarifies the boundary of trajectory optimization, into which coasting regimes can be added for energy consumption reduction with running time addition. Meanwhile, the trajectory is divided into several subintervals for better determining the starting points of coasting regimes. Finally, a coasting control algorithm is designed to calculate the energy-saving train trajectory meeting the constraint of pregiven running time. Coasting regimes are added into the subintervals with high energy consumption reduction efficiency, which means that more energy consumption can be reduced with the same running time addition. Specially, the proposed coasting control algorithm can be applied to the energy-saving train control problem for multisection, and then the energy-saving train trajectory and running time schemes can be generated simultaneously.

The rest of this paper is organized as follows. In Section 2, we describe the energy-saving train control problem. Then, we propose the solution methodology for the optimal control problem based on coasting control in Section 3. In Section 4, we give two numerical examples, i.e., a single-section case and a multisection case based on one of the Beijing metro lines, to demonstrate the effectiveness and efficiency of the proposed approaches. We conclude this paper in Section 5.

2. Problem Formulation

2.1. Definition of Symbols. For a better understanding of our paper, we define the necessary notations and parameters for the energy-efficient train control problem in Table 1.

2.2. Train Dynamics Modeling. By considering the train traction, braking force, running basic resistance, and line resistance, the dynamics model of train motion can be formulated as follows [24].

$$v\frac{\mathrm{d}v}{\mathrm{d}x} = \frac{F - B - W_0 - W_l}{M(1+\gamma)}, \qquad \frac{\mathrm{d}t}{\mathrm{d}x} = \frac{1}{v}. \tag{1}$$

The train traction force F can be expressed as follows:

$$F = \mu_f F_m(v), \tag{2}$$

where F_m is the maximum traction force, which is dependent on the train characteristics and speed; μ_f is the

Table 1: Notations and parameters.

Index	Description
i	Index of station
j	Index of subinterval
k	Index of discretizing point
Parameters	**Description**
M	Train mass (t)
γ	Rotary mass coefficient
η	Efficiency of train motor
V	Speed restriction (m/s)
S_i	Position of station i (m)
$T_{i,i'}$	Running time from station i to station i' (s)
W_0	Train running basic resistance (kN)
W_l	Line resistance (kN)
F_m	Maximum train traction force (kN)
B_m	Maximum train braking force (kN)
State variables	**Description**
x	Train running position (m)
t	Train running time (s)
v	Train running speed (m/s)
F	Train traction force (kN)
B	Train braking force (kN)
Decision variables	**Description**
μ_f	Train traction force adjustment coefficient
μ_b	Train braking force adjustment coefficient

traction force adjustment coefficient, $\mu_f \in [0,1]$. Similarly, the train braking force B can be expressed as follows:

$$B = \mu_b B_m(v), \tag{3}$$

where B_m is the maximum braking force, which is dependent on the train characteristics and speed; μ_b is the braking force adjustment coefficient, $\mu_b \in [0,1]$. Based on the adjustment coefficients μ_f and μ_b, the corresponding train control regimes can be described as shown in Table 2.

The train running basic resistance can be formulated based on the Davis equation.

$$W_0(v) = M\left(a_1 + a_2 v + a_3 v^2\right), \tag{4}$$

where a_1, a_2, and a_3 are nonnegative coefficients. Moreover, the line resistance, caused by track slope, can be calculated as follows:

$$W_l(x) = Mg\sin(\theta(x)), \tag{5}$$

where $\theta(x)$ is the track slope at position x.

In addition, we adopt distance-based modelling to generate the energy-efficient train control strategy. The whole section $[S_i, S_{i+1}]$ is divided into K subsections, $K\Delta x = S_{i+1} - S_i$, as shown in Figure 1. The postions of discretizing points are denoted as $\{x_1, \ldots, x_{K+1}\}$, and satisfy:

$$x_{k+1} = x_k + \Delta x. \tag{6}$$

Meanwhile, the train running speed of discretizing points are denoted as $\{v_1, \ldots, v_{K+1}\}$. The relationship between v_{k+1} and v_k can be described as follows:

$$v_{k+1}^2 - v_k^2 = 2a_k\Delta x. \tag{7}$$

Based on equation (7), v_{k+1} can be calculated as follows:

$$v_{k+1} = \sqrt{v_k^2 + 2a_k\Delta x}, \tag{8}$$

where a_k is the train acceleration in subsection $[x_k, x_{k+1}]$, which can be calculated as follows:

$$a_k = \frac{\mu_{f,k}F_{m,k} - \mu_{b,k}B_{m,k} - W_{0,k} - W_{l,k}}{M(1+\gamma)}, \tag{9}$$

where F_k, B_k, $W_{0,k}$, and $W_{l,k}$ are the train traction force, braking force, basic resistance, and line resistance in subsection $[x_k, x_{k+1}]$, respectively. Finally, the train running times of discretizing points are denoted as $\{t_1, \ldots, t_{K+1}\}$, and satisfy:

$$t_{k+1} = \frac{2\Delta x}{v_k + v_{k+1}} + t_k. \tag{10}$$

2.3. Energy-Saving Train Control Model for Single Section. In this section, based on the distance-based modelling, we introduce the energy-saving train control model for a single-section $[S_i, S_{i+1}]$. The train running process from station i to station $i+1$ is analyzed, which is divided into K subsection. First, the train traction energy consumption is introduced. At each subsection, the train traction energy consumption can be calculated as follows:

$$E_k = F_k\frac{\Delta x}{\eta}. \tag{11}$$

Then, the cost function of energy-saving train control problem for the single-section can be described as follows:

$$J_{i,i+1} = \sum_{k=1}^{K} E_k. \tag{12}$$

For the train running process from station i to station $i+1$, the following constraints should be considered. The train speed at station i and station $i+1$ are equal to zero:

$$v_{K+1} = v_1 = 0. \tag{13}$$

To keep the safe operation, the train speed must be less than the speed restrictions:

$$0 \le v_k \le V_k. \tag{14}$$

To keep the punctual operation, the pre-given running time should be satisfied:

$$t_{K+1} - t_1 = T_{i,i+1}. \tag{15}$$

Then, the energy-saving train control problem for the single-section $[S_i, S_{i+1}]$ can be stated as follows.

$$\begin{cases} \min\ J_{i,i+1} = \sum_{k=1}^{K} E_k, \\ \text{s.t. } \mu_{f,k}, \mu_{b,k} \in [0,1], \\ \text{Equations (6) to (10), and (13) to (15).} \end{cases} \tag{16}$$

Table 2: Description of control regimes based on μ_f and μ_b.

Regime	Description
MA	$\mu_f = 1$ and $\mu_b = 0$
CR	$\mu_f \in (0,1)$ and $\mu_b = 0$ or $\mu_f = 0$ and $\mu_b \in (0,1)$
CO	$\mu_f = 0$ and $\mu_b = 0$
MB	$\mu_f = 0$ and $\mu_b = 1$

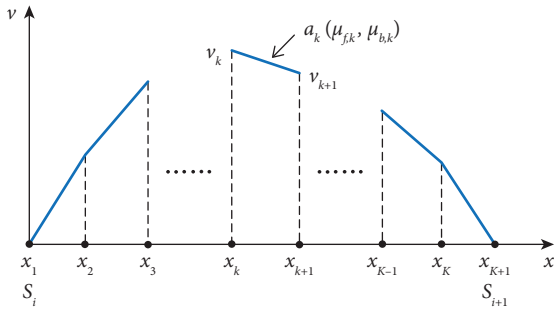

Figure 1: The illustration of the section discretizing.

By solving the abovementioned optimal control problem 17, optimal control actions $\{\mu^*_{f,1}, \ldots, \mu^*_{f,K}\}$ and $\{\mu^*_{b,1}, \ldots, \mu^*_{b,K}\}$ can be obtained to generate the energy-efficient train trajectory.

2.4. Energy-Saving Train Control Model for Multisection. Based on the energy-saving train control model for a single-section, the model for multisection is built in this section. Considering the train running process from station i to station $i + I$, $I \geq 2$, the whole section $[S_i, S_{i+I}]$ is also divided into K subsections, $K\Delta x = S_{i+I} - S_i$, as shown in Figure 2. A set $\{\kappa_i, \ldots, \kappa_{i+I}\}$ is introduced to represent the indexes of the discretizing points that overlap with the station positions $\{S_i, \ldots, S_{i+I}\}$, which can be described as follows:

$$x_{\kappa_{i'}} = S_{i'}, \forall i' \in \{i, \ldots, i + I\}. \tag{17}$$

Considering the middle stations in the multisection running process, the constraint 14 that limits the train speed at stations should be rewritten as follows:

$$v_k = 0, \forall k \in \{\kappa_i, \ldots, \kappa_{i+I}\}. \tag{18}$$

Meanwhile, the punctuality constraint 16 should be rewritten as follows:

$$t_{K+1} - t_1 = T_{i,i+I}, \tag{19}$$

where only the total running time from station i to station $i + I$ is limited, the running times of each section are unlimited, which can be adjusted in the optimization process.

Then, the energy-saving train optimal control problem for the multisection $[S_i, S_{i+I}]$ can be stated as follows.

$$\begin{cases} \min J_{i,i+I} = \sum_{k=1}^{K} E_k, \text{s.t.} \quad \mu_{f,k}, \mu_{b,k} \in [0,1]. \end{cases} \tag{20}$$

Equations (6–10), (14), (18), and (19).

The energy-efficient train trajectory from station i to station $i + I$ can be obtained by solving the optimal train control problem 21. Meanwhile, the energy-efficient distribution of running time $\{T^*_{i,i+1}, \ldots, T^*_{i+I-1,i+I}\}$ can be generated, which can be calculated as follows:

$$T^*_{i',i'+1} = t^*_{k_{i'+1}} - t^*_{k_{i'}}, \forall i' \in \{i, \ldots, i + I - 1\}. \tag{21}$$

3. Coasting Control Algorithm

Typically, the pre-given running times are not those associated with the minimum running times (flat-out trajectories), but they include running time supplements to be able to recover delays when necessary or fulfil running at a lower speed with less energy consumption [25]. Coasting control is an economical approach to balance running time and energy consumption in train operation [21]. In this section, we introduce a coasting control algorithm to calculate the energy-saving trajectory. First, the calculation of flat-out trajectory is introduced to generate the minimum running time. Then, the subinterval is proposed to divide the whole running process into sections, in which coasting regimes can be added. Finally, coasting points are determined based on the principle that adding coasting regimes to subintervals with high energy-saving effects.

3.1. Flat-Out Trajectory. Under flat-out running, a train is travelling close to speed restrictions. As shown in Figure 3, when the train speed is less than the speed restriction, MA regime is applied to speed up; when the speed is close to the speed restriction, CR regime is applied to keep the train running at high-speed; MB regime is applied for low-speed restrictions and stopping. Thus, there is no coasting regime in the trajectory. This kind of trajectory is defined as the flat-out trajectory [23], which can be calculated as shown in Algorithm 1 and Figure 4. In the flat-out trajectory calculation algorithm, train control regimes are determined based on the relationship between the train running speed and speed restriction. Under the constraints of safety and train characteristics, the algorithm keeps the train running speed as close to or within speed restrictions.

Based on the flat-out trajectory calculation algorithm, the minimum running time can be calculated as follows:

$$T^{\text{flat}}_{i,i'} = t^{\text{flat}}_{K+1} - t_1, \tag{22}$$

where $T^{\text{flat}}_{i,i'}$ is the minimum running time from station i to station i'; t^{flat}_{K+1} is the train running time at the ending point (station i') of the flat-out trajectory.

3.2. Definition of Subinterval. Based on the flat-out trajectory, coasting regimes can be added into the control sequence, as shown in Figure 3. For better determining the starting points of coasting regimes, the whole running process is divided into several subintervals. The subinterval means the train running process starts with an accelerating regime and ends with a braking regime. Meanwhile, there

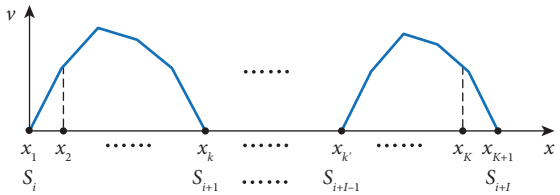

FIGURE 2: The illustration of multisection discretizing.

can only be one accelerating phase and one braking phase in a subinterval, which makes the accelerating phase and its subsequent adjacent braking phase form a subinterval. As shown in Figure 3, the first subinterval begins with a MA (accelerating) regime and ends with an MB (braking) regime, and the MA and MB regimes are adjacent. Four important points (x_{a_j}, x_{c_j}, x_{d_j}, and x_{b_j}) are introduced to describe the subinterval j, as shown in Figure 5.

(1) x_{a_j}: the beginning position of the accelerating regime
(2) x_{c_j}: the beginning position of the coasting regime
(3) x_{d_j}: the ending position of the coasting regime
(4) x_{b_j}: the ending position of the braking regime.

Without coasting regime addition, x_{c_j} is equal to the ending position of the accelerating regime, and x_{d_j} is equal to the starting position of the braking regime. As the duration of the coasting regime increases, the beginning position of the coasting regime x_{c_j} moves to x_{a_j}, and the ending position of the coasting regime x_{d_j} moves to x_{b_j}. Specially, when $x_{d_j} = x_{b_j}$ or $x_{c_j} = x_{a_j}$, the duration of the coasting regime cannot increase, as shown in Figure 6. For two adjacent subintervals, if $x_{d_j} = x_{b_j}$ in the first subinterval and $x_{c,j+1} = x_{a,j+1}$ in the second subinterval, then these two adjacent subintervals can be merged as a new subinterval, as shown in Figure 6. Specially, the beginning position of the accelerating regime and the beginning position of the coasting regime of the new subinterval are equal to those of subinterval j, the ending position of the coasting regime and the ending position of the braking regime of the new subinterval are equal to those of subinterval $j + 1$.

3.3. Coasting Points Determination. Due to the line characteristics (like speed restrictions and track slope), the flat-out trajectory consists of several subintervals, as shown in Figure 3. We propose a coasting control algorithm to distribute the running time supplements to subintervals. The distribution criterion distributes the running time supplements to subintervals where energy consumption can be reduced more significantly. To evaluate the efficiency of energy consumption reduction, indicator ρ_j is introduced:

$$\rho_j = -\frac{\Delta E_j}{\Delta T_j} = -\frac{\sum_{k=a_j}^{b_j} (E_k' - E_k)}{t_{b_j}' - t_{b_j}}, \tag{23}$$

where ρ_j is the energy-saving effect of subinterval j; $\Delta E_j = \sum_{k=a_j}^{b_j} (E_k' - E_k)$ is the energy consumption change value after adding the coasting regime of subinterval j; $\Delta T_j = t_{b_j}' - t_{b_j}$ is

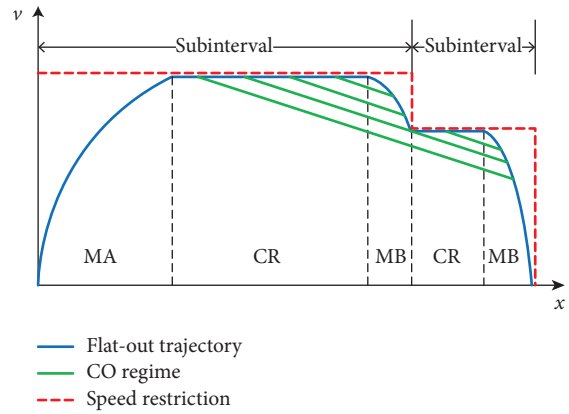

FIGURE 3: The illustration of flat-out trajectory and coasting regimes addition.

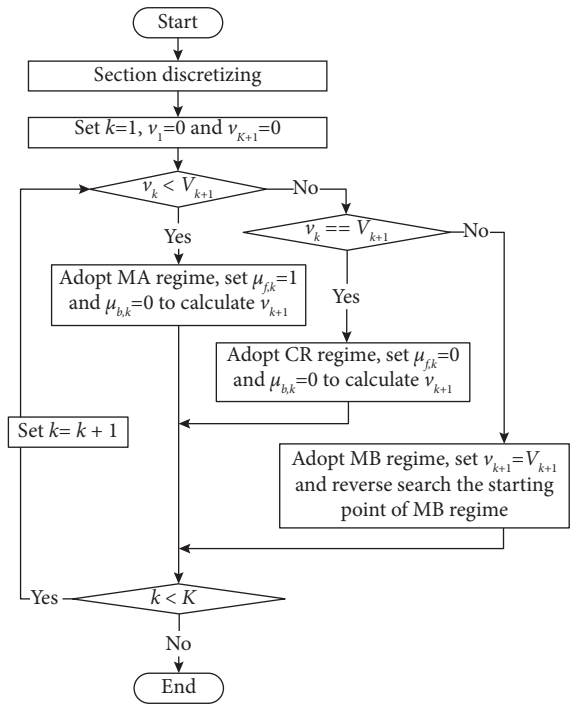

FIGURE 4: The illustration of the flat-out trajectory calculation algorithm.

the running time change value after adding the coasting regime of subinterval j. In addition, the larger ρ_j means that the more energy consumption can be reduced in subinterval j with the same running time supplement. Considering the whole running process from station i to station i', the limited running time supplement should be distributed to the subinterval with the maximum ρ_j for energy saving. Based on this, the coasting control algorithm is proposed:

Specially, for the multisection running process, the proposed coasting control algorithm is effective. Due to the multisection running process, the speed limit constraints of middle stations need to be considered additionally. First, the multisection running process will be divided into several subintervals with the same number of sections, as shown in

(1) Divide the whole running section into K subsections. Set $k = 1$, $v_1 = 0$, and $V_{K+1} = 0$.
(2) **while** $k \le K$ **do**
(3) **if** $v_k < V_{k+1}$ (MA regime) **then**
(4) Set $\mu_{f,k} = 1$ and $\mu_{b,k} = 0$ to calculate v_{k+1}.
(5) **else if** $v_k = V_{k+1}$ (CR regime) **then**
(6) Set $v_{k+1} = Vk + 1$ to calculate $\mu_{f,k}$ and $\mu_{b,k}$.
(7) **else if** $v_k > V_{k+1}$ (MB regime) **then**
(8) Set $v_{k+1} = V_{k+1}$ and $k' = k + 1$. Then, reverse search the starting point of MB regime (k'):
(9) **repeat**
(10) Set $\mu'_{f,k'} = 0$ and $\mu'_{b,k'} = 1$ to calculate $v'_{k'-1}$. Since , set $k' = k' - 1$.
(11) **until** $v'_{k'} = v_k$
(12) Set $\mu_{f,\kappa} = \mu'_{f,\kappa}$, $\mu_{b,\kappa} = \mu'_{b,\kappa}$, and $v_\kappa = v'_\kappa$, for $\kappa \in \{k', k' + 1, \ldots, k\}$
(13) **end if**
(14) $k = k + 1$
(15) **end while**

Algorithm 1:Flat-out trajectory calculation.

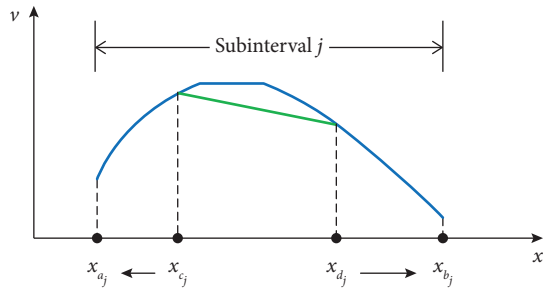

Figure 5: The illustration of the subinterval.

Figure 2. Then, each section will be divided into several subintervals based on the definition of the subinterval. Thus, the running time supplement can be distributed for the multisection running process as described in the Algorithm 2, to generate energy-saving running time schemes and related trajectories.

4. Numerical Examples

In this section, we present two numerical examples to demonstrate the performance of the proposed energy-efficient coasting control algorithm. The first example solves the energy-saving train control problem along the single-section, in which only the energy-saving train trajectories are optimized. The second one involves the energy-saving train control problem along the multisection, in which both the energy-saving train trajectories and running time scheme are optimized.

All examples are based on the data of one of the Beijing metro lines. The speed restrictions and track gradient between station 1 and station 5 are shown in Figure 7. The parameters of the running train are listed in Table 3. The maximum train traction force, the maximum braking force, and running basic resistance are given as follows:

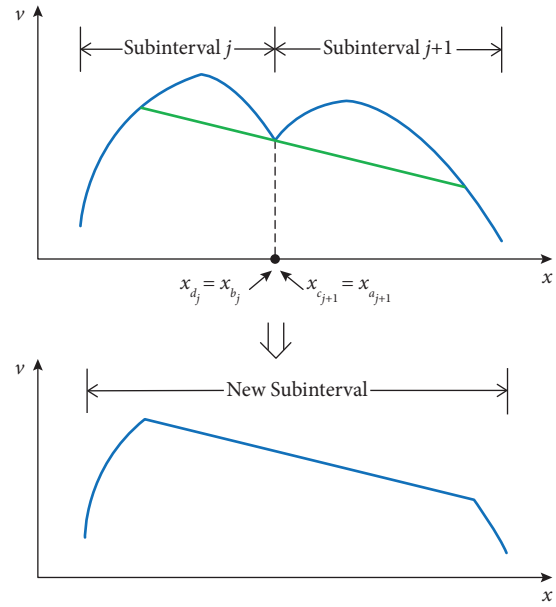

Figure 6: The illustration of the subinterval merging.

$$F_m(v) = \begin{cases} 200 & 0 \le v \le 15.28, \\ \dfrac{46682}{v^2} & 15.28 < v < \le 22.22, \end{cases} \tag{24}$$

$$B_m(v) = 159.6 \quad 0 \le v \le 22.22,$$

$$W_0(v) = 216\left(4.5024 + 0.1089v + 0.0108v^2\right).$$

Examples are tested under the MATLAB environment on a personal computer with Intel Core i5 2.30 GHz CPU and 8 GB RAM.

(1) Divide the whole running section into K subsections. Calculate the flat-out trajectory based on the Algorithm 1. Initialize the running time supplement $T_{i,i'}^{\sup} = T_{i,i'} - T_{i,i'}^{\text{flat}}$. Divide the whole running section into J subintervals based on the definition of the subinterval.
(2) **while** $T_{i,i'}^{\sup} \geq 0$ **do**
(3) **for** $j = 1$ to J **do**
(4) Move the beginning position of coasting regime x_{c_j} with step Δx backward temporarily, $c_j' = c_j - 1$. Calculate the ending position of coasting regime x_{d_j}' with $\mu_{f,k'} = \mu_{b,k'} = 0$, for $k' \in \{c_j', \ldots, d_j'\}$. Calculate ρ_j based on the temporarily modified trajectory.
(5) **end for**
(6) Determine the subinterval j' with maximum ρ_j', $\rho_j' \geq \rho_j$, for $j \in \{1, \ldots, J\}$. Update $c_{j'} = c_{j'}'$, $d_{j'} = d_{j'}'$, and $T_{i,i'}^{\sup} = T_{i,i'}^{\sup} - \Delta T_{j'}$.
(7) **for** $j = 1$ to $J - 1$ **do**
(8) **if** $x_{d_j} = x_{b_j}$ and $x_{c,j+1} = x_{a,j+1}$ **then**
(9) Merge the subinterval j and subinterval $j + 1$. Update $J = J - 1$.
(10) **end if**
(11) **end for**
(12) **end while**

Algorithm 2: Coating control algorithm.

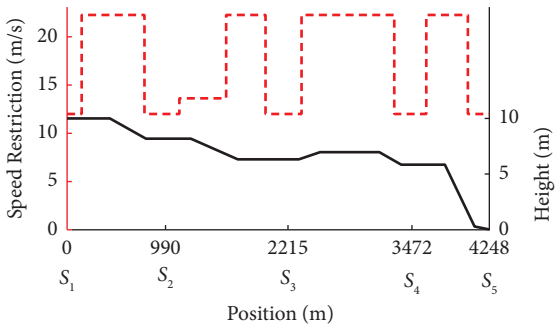

Figure 7: The illustration of speed restriction and gradient changing.

Table 3: Train parameters.

Parameters	Value
Train mass, M	216 (t)
Rotary mass coefficient, γ	0.08
Efficiency of train motor, η	0.85

4.1. Example 1: Scenarios of Single-Section Running. In this example, the train trajectories of sections $[S_1, S_2]$, $[S_2, S_3]$, $[S_3, S_4]$, and $[S_4, S_5]$ were analyzed. Four different types of trajectories are compared to verify the performance of the proposed coasting control algorithm for single-section train control. More details about these four trajectories are as follows:

(1) T-Pra: Practical trajectory obtained from the equipped ATO systems
(2) T-flat: Flat-out trajectory with a minimum running time and maximum energy consumption, which can be calculated based on the Algorithm 1;
(3) T-CC: Optimal trajectory calculated based on the proposed coasting control algorithm (Algorithm 2);
(4) T-DP: Optimal trajectory calculated based on dynamic programming, more details can be seen in Appendix.

Specially, the DP algorithm is introduced to compare with the coasting control algorithm, aiming to demonstrate its effect. Δx is set to be 1 m in the coasting control algorithm, and then the running processes of four sections are divided into 990, 1225, 1257, and 776 subsections, respectively. Meanwhile, Δx is also set to be 1 m in the DP algorithm, and Δv is set to be 0.02 m/s. For four different types of trajectories, the trajectories and control commands are shown in Figure 8, and the performance is shown in Table 4.

As shown in Figure 8, T-flat is keeping close to the speed restrictions in the running process, in which there is no coasting regime. The T-flat corresponds to the maximum energy consumption and the minimum running time in each section, as shown in Table 4. For T-Pra, accelerating regimes are applied to reach a high speed at the beginning of the running process, then braking regimes are applied for the low-speed restriction and train stops. Compared to T-CC and T-DP, we can observe that fewer coasting regimes are applied in T-Pra, as shown in Figure 8. For T-CC and T-DP, MA regimes are applied at the beginning of the running process and MB regimes are applied at the stopping process. This kind of strategy avoids the train from staying in low-speed phases and wasting running time supplements. Considering the running time constraint, there will be more time for train coasting to reduce energy consumption. Therefore, as shown in Table 4, the comparison results of single-section running with the same running times show that T-CC can achieve 26.16%, 37.51%, 12.12% and 35.31%, energy-saving for sections $[S_1, S_2]$, $[S_2, S_3]$, $[S_3, S_4]$, and $[S_4, S_5]$, respectively, in comparison to T-Pra. Meanwhile, T-DP can achieve 26.46%, 37.35%, 12.40%, and 36.19% energy-saving for four sections, respectively. The little deviations in energy-saving performance between T-CC and

Table 4: The comparison of different trajectories and running time distribution schemes from S_1 to S_5.

Section index	$[S_1, S_2]$		$[S_2, S_3]$		$[S_3, S_4]$		$[S_4, S_5]$		$[S_1, S_5]$ (total)	
	$T_{1,2}$ (s)	$J_{1,2}$ (kWh)	$T_{2,3}$ (s)	$J_{2,3}$ (kWh)	$T_{3,3}$ (s)	$J_{3,4}$ (kWh)	$T_{4,5}$ (s)	$J_{4,5}$ (kWh)	$T_{1,5}$ (s)	$J_{1,5}$ (kWh)
T-flat	80.97	20.38	103.51	17.63	91.59	22.24	71.72	15.94	347.79	76.19
T-Pra	88.40	13.15	109.80	12.45	100.90	14.44	79.80	9.09	378.90	49.13
T-DP	88.31	9.67	109.72	7.80	100.90	12.65	79.78	5.80	378.71	35.92
T-CC	88.31	9.71	109.77	7.78	100.82	12.69	79.78	5.88	378.68	36.06
T-CC with optimal running times	89.46	9.12	110.17	7.58	101.03	12.59	78.22	6.53	378.88	35.83

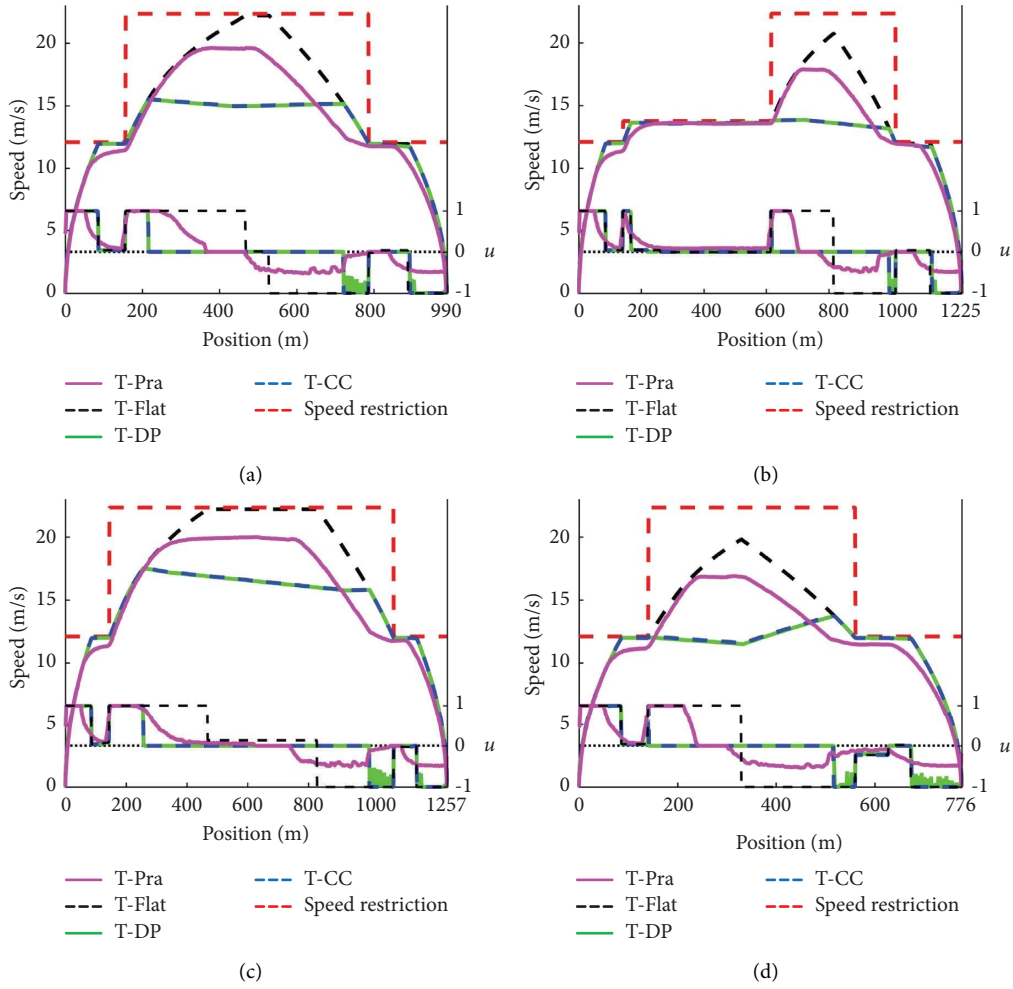

Figure 8: The illustration of trajectories and control commands μ for four sections. (a) Section $[S_1, S_2]$. (b) Section $[S_2, S_3]$. (c) Section $[S_3, S_4]$. (d) Section $[S_4, S_5]$.

T-DP might come from the speed discretization in the DP algorithm and the small differences in running times.

In terms of computation time, the average computation times of T-CC and T-DP for four sections are 1.1 s and 261.9 s, respectively. This means that, for single-section train control, applying the coasting control algorithm can achieve the similar energy-saving effect as the DP algorithm with less computation time. Specially, the computation time of the coasting control algorith m can reach a 10 ms level when running in a *C* environment.

In addition, to verify the feasibility of the coasting control algorithm, the optimal trajectories with different running times in sections $[S_1, S_2]$ are analyzed. T-CC with different running times is shown in Figure 9, and the corresponding running time supplements of each subinterval and energy consumption are shown in Table 5. First, the whole running process from S_1 to S_2 is divided into two subintervals due to the low-speed restriction, as shown in Figure 9. As the running time increases, running time supplements are added into the subintervals with coasting

Table 5: The comparison of T-CC with different running times from S_1 to S_2.

Pre-given running time (s)	Running time supplement in subinterval 1 (s)	Running time supplement in subinterval 2 (s)	Energy consumption (kWh)
80.97	0	0	20.38
81.97 (+1)	0.96	0.03	16.16
82.97 (+2)	1.91	0.07	14.34
85.97 (+5)	4.86	0.08	11.23
90.97 (+10)	9.90	0.08	8.47
100.97 (+20)	20.318 (merged subinterval)		5.72
110.97 (+30)	29.672 (merged subinterval)		4.58

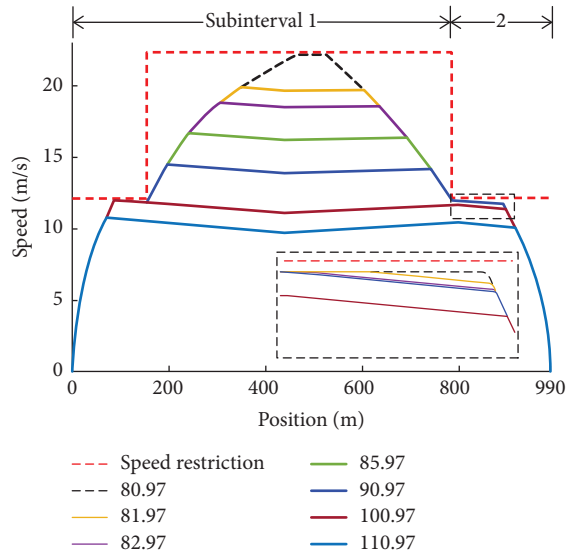

Figure 9: The illustration of T-CC with different running times from S_1 to S_2.

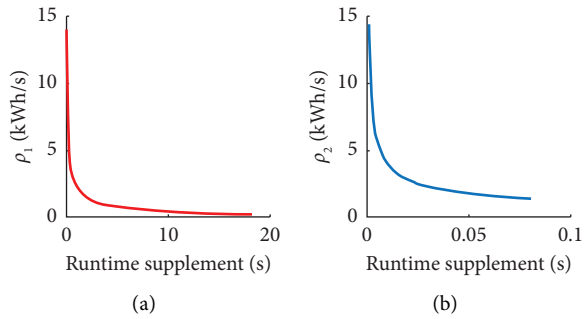

Figure 10: The illustration of energy-saving effects ρ for the subinterval 1 (a) and 2 (b)

duration addition. In addition, the energy-saving effects ρ of the subintervals 1 and 2, as shown in Figure 10, guide the distribution of running time supplements. Running time supplement is added to the subinterval with the larger ρ for more energy consumption reduction. Specially, the maximum running time supplement of the subinterval 2 is 0.08 s. When the running time supplement of subinterval reaches the maximum one, there is no room for coasting regime addition, like the cases 85.97 s and 90.97 s. Since, when the running time supplement is large enough, two subintervals are merged into one subinterval, as in the cases 100.97 s and 110.97 s in Figure 9.

4.2. Example 2: Scenarios of Multisection Running. In this example, we optimize the train trajectories and related to running time schemes for the running process from S_1 to S_2, based on the proposed coasting control algorithm. T-CC with optimal running times is compared with the other trajectories with pre-given running times, to verify the performance of the coasting control algorithm for the multisection train control. T-CC with optimal running times represents the optimal trajectories calculated based on the optimal model 21, in which trajectories and running times for the multisection running process are optimized integratedly. T-CC with optimal running times and T-CC with

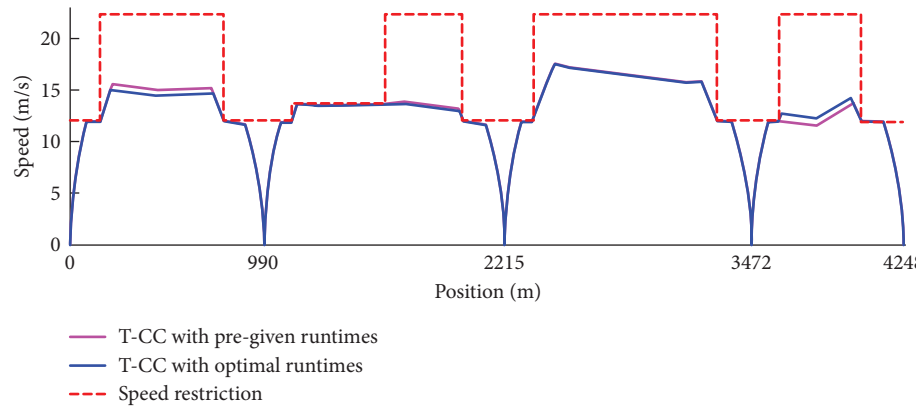

Figure 11: The illustration of T-CC with optimal running times and T-CC with pre-given running times from S_1 to S_5.

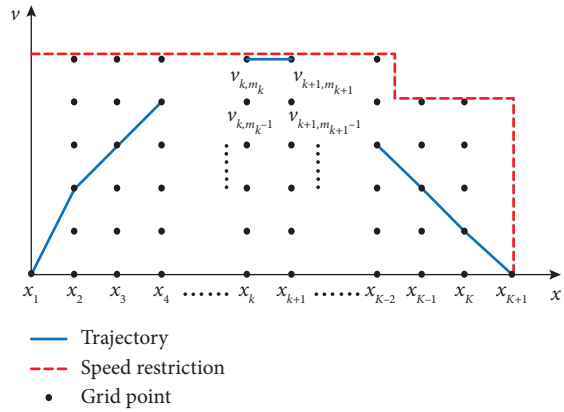

Figure 12: The illustration of the speed-distance network.

pre-given running times are compared in Figure 11, and the train running times and energy consumption of each section for T-CC with optimal running times are shown in Table 4.

As shown in Table 4, by comparing T-CC with pre-given running times and T-CC with optimal running times, we can observe that the running times of each section are different in these two plans. Due to the change in running times, there are also differences in the trajectories. As shown in Figure 11, with running time addition, more coasting regimes are added into the trajectories in sections $[S_1, S_2]$, $[S_2, S_3]$, and $[S_3, S_4]$ of T-CC with optimal running times. Meanwhile, the energy consumption in these three sections can be reduced in comparison to those of T-CC with pre-given running times. On the other hand, with running time reduction in the sections $[S_4, S_5]$, the energy consumption of T-CC with optimal running times in this section is larger than it of T-CC with the pre-given running times. In terms of the whole running process, for T-CC with optimal running times, the total energy consumption can be reduced from 36.06 kWh to 35.83 kWh compared to T-CC with pre-given running times.

5. Conclusion

In this paper, we studied the optimal train control problem to reduce energy consumption. Combining the operational constraints and energy-saving objective, we developed distance-based train trajectory optimization models for the single-section and multi-section operations. A coasting control algorithm was proposed to generate the energy-efficient trajectories, in which the coasting control regime points were determined according to the energy-saving effect.

Numerical examples based on one of the Beijing metro lines were implemented in two different cases, i.e., single-section and multisection operation, to demonstrate the performance of the proposed coasting control algorithm. The computational results showed that, by applying the coasting control algorithm, the energy consumption of single-section operation can be reduced effectively by around 12.12% to 37.35% in comparison to the practical trajectories obtained from equipped ATO systems. Meanwhile, the coasting control algorithm was compared with the DP algorithm; the former can achieve a similar energy-saving performance in shorter computation times. For the multisection operation, the proposed coasting control algorithm can generate energy-saving running time schemes and related trajectories by optimizing the whole running process integratedly.

Our future research will focus on the online train control problem to deal with the dynamic situations, like temporary speed restrictions. This paper only deals with the offline train

trajectory optimization problem. However, the train trajectories will be adjusted in real-time operation to keep safe and punctual operations.

Appendix

Dynamic programming for energy-saving train control

Dynamic programming is introduced in this section, which has been widely applied in the energy-saving train control problem [13–17]. First, the whole speed-distance space is discretized into different stages $\{1, 2, \ldots, K, K+1\}$, over which all the possible speed variations can be represented as a variety of links between different grid points [16], as shown in Figure 12. φ_k is introduced to represent the state set of stage k:

$$\varphi_k = (x_k, v_k) \in \left\{\left(x_k, v_{k,1}\right), \ldots, \left(x_k, v_{k,m_k}\right)\right\}, \quad (A.1)$$

where m_k is determined by the speed restriction V_k, $m_k \Delta v = V_k$. According to the constraint 14, the initial state and the final state should be equal to zero, which can be described as follows:

$$\varphi_1 = (0, 0), \varphi_{K+1} = (x_{K+1}, 0). \quad (A.2)$$

Introducing $p(v_k, v_{k+1})$ as the indicator function from stage $k+1$ to stage k, which can be calculated as follows:

$$p(v_k, v_{k+1}) = E_k + \lambda \tau_k, \quad (A.3)$$

where E_k can be calculated based on equation (12); τ_k is equal to $2\Delta x/(v_k + v_{k+1})$; λ is a weight coefficient to balance energy consumption and running time. In addition, the cumulative indicator P_k will be calculated:

$$P_k = \min\{p(v_k, v_{k+1}) + \min(pv_{k+1}, v_{K+1}]\}. \quad (A.4)$$

The dynamic programming process of the energy-efficient train control includes two steps: backward calculation and forward search [13]. In the first step, the optimal policy on the grid point of state space is determined and recorded from $k = K+1$ to $k = 1$. Then, an optimal trajectory can be created by searching forward from $k = 1$ to $k = K+1$ based on the optimal policy.

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant nos. U1934221 and 62003283.

References

[1] G. M. Scheepmaker, R. M. Goverde, and L. G. Kroon, "Review of energy-efficient train control and timetabling," *European Journal of Operational Research*, vol. 257, no. 2, pp. 355–376, 2017.

[2] A. González-Gil, R. Palacin, P. Batty, and J. P. Powell, "A systems approach to reduce urban rail energy consumption," *Energy Conversion and Management*, vol. 80, pp. 509–524, 2014.

[3] K. Ishikawa, "Application of optimization theory for bounded state variable problems to the operation of trains," *Bull. JSME*, vol. 11, no. 47, pp. 857–865, 1968.

[4] P. J. Pudney and P. Howlett, "Optimal driving strategies for a train journey with speed limits," *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics*, vol. 36, no. 1, pp. 38–49, 1994.

[5] P. G. Howlett, P. J. Pudney, and X. Vu, "Local energy minimization in optimal train control," *Automatica*, vol. 45, no. 11, pp. 2692–2698, 2009.

[6] A. R. Albrecht, P. G. Howlett, P. J. Pudney, and X. Vu, "Energy-efficient train control: from local convexity to global optimization and uniqueness," *Automatica*, vol. 49, no. 10, pp. 3072–3078, 2013.

[7] G. M. Scheepmaker and R. M. Goverde, "The interplay between energy-efficient train control and scheduled running time supplements," *Journal of Rail Transport Planning & Management*, vol. 5, no. 4, pp. 225–239, 2015.

[8] A. Albrecht, P. Howlett, P. Pudney, X. Vu, and P. Zhou, "The key principles of optimal train control—Part 1: formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points," *Transportation Research Part B: Methodological*, vol. 94, pp. 482–508, 2016.

[9] A. Albrecht, P. Howlett, P. Pudney, X. Vu, and P. Zhou, "The key principles of optimal train control—Part 2: existence of an optimal strategy, the local energy minimization principle, uniqueness, computational techniques," *Transportation Research Part B: Methodological*, vol. 94, pp. 509–538, 2016.

[10] W. Zhong, S. Li, H. Xu, and W. Zhang, "On-line train speed profile generation of high-speed railway with energy-saving: a model predictive control method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 1–12, 2020.

[11] Y. Wang, B. De Schutter, T. J. van den Boom, and B. Ning, "Optimal trajectory planning for trains under operational constraints using mixed integer linear programming," *IFAC Proceedings Volumes*, vol. 45, pp. 13–18, 2012.

[12] Y. Wang, B. De Schutter, T. J. Van Den Boom, and B. Ning, "Optimal trajectory planning for trains under fixed and moving signaling systems using mixed integer linear programming," *Control Engineering Practice*, vol. 22, no. 1, pp. 44–56, 2014.

[13] H. Ko, T. Koseki, and M. Miyatake, "Application of dynamic programming to optimization of running profile of a train," *Computers in Railways IX*, pp. 103–112, WIT Press, Ashurst london, UK, 2004.

[14] S. Lu, S. Hillmansen, T. K. Ho, and C. Roberts, "Single-train trajectory optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 743–750, 2013.

[15] J. T. Haahr, D. Pisinger, and M. Sabbaghian, "A dynamic programming approach for optimizing train speed profiles with speed restrictions and passage points," *Transportation Research Part B: Methodological*, vol. 99, pp. 167–182, 2017.

[16] L. Wang, L. Yang, Z. Gao, and Y. Huang, "Robust train speed trajectory optimization: a stochastic constrained shortest path approach," *Frontiers of Engineering Management*, vol. 4, no. 4, p. 408, 2017.

[17] S. Ichikawa and M. Miyatake, "Energy efficient train trajectory in the railway system with moving block signaling scheme," *IEEJ Journal of Industry Applications*, vol. 8, no. 4, pp. 586–591, 2019.

[18] P. Wang and R. M. Goverde, "Multiple-phase train trajectory optimization with signalling and operational constraints," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 255–275, 2016.

[19] P. Wang and R. M. Goverde, "Multi-train trajectory optimization for energy efficiency and delay recovery on single-track railway lines," *Transportation Research Part B: Methodological*, vol. 105, pp. 340–361, 2017.

[20] C. S. Chang and S. S. Sim, "Optimising train movements through coast control using genetic algorithms," *IEE Proceedings - Electric Power Applications*, vol. 144, no. 1, pp. 65–72, 1997.

[21] K. K. Wong and T. K. Ho, "Dynamic coast control of train movement with genetic algorithm," *International Journal of Systems Science*, vol. 35, no. 13-14, pp. 835–846, 2004.

[22] Y. Bocharnikov, A. M. Tobias, C. Roberts, S. Hillmansen, and C. Goodman, "Optimal driving strategy for traction energy saving on DC suburban railways," *IET Electric Power Applications*, vol. 1, no. 5, pp. 675–682, 2007.

[23] B. Jin, Q. Fang, Q. Wang, P. Sun, and X. Feng, "Energy-efficient train control in urban rail transit: multi-train dynamic cooperation based on train-to-train communication," in *Proceedings of the 2021 IEEE Intelligent Vehicles Symposium*, pp. 309–314, Nagoya, Japan, July 2021.

[24] Y. Song, W. Song, and W. Song, "A novel dual speed-curve optimization based approach for energy-saving operation of high-speed trains," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1564–1575, 2016.

[25] A. Fernández-Rodríguez, A. P. Cucala, and A. Fernández-Cardador, "An eco-driving algorithm for interoperable automatic train operation," *Applied Sciences*, vol. 10, no. 21, pp. 7705–7729, 2020.