Deep Deterministic Policy Gradient for High-Speed Train Trajectory Optimization

Ning, Lingbin; Zhou, Min; Hou, Zhuopu; Goverde, Rob M.P.; Wang, Fei Yue; Dong, Hairong

DOI
10.1109/TITS.2021.3105380

Publication date
2022

Document Version
Final published version

Published in
IEEE Transactions on Intelligent Transportation Systems

Deep Deterministic Policy Gradient for High-Speed Train Trajectory Optimization

Lingbin Ning, Min Zhou, *Member, IEEE*, Zhuopu Hou, Rob M.P. Goverde, *Member, IEEE*, Fei-Yue Wang, *Fellow, IEEE*, and Hairong Dong, *Senior Member, IEEE*

***Abstract*— This paper proposes a novel train trajectory optimization approach for high-speed railways. We restrict our attention to single train operation scenarios with different scheduled/rescheduled running times aiming at generating optimal train recommended trajectories in real time, which can ensure punctuality and energy efficiency of train operation. A learning-based approach deep deterministic policy gradient (DDPG) is designed to generate optimal train trajectories based on the offline training from the interaction between the agent and the trajectory simulation environment. An allocating running time and selecting operation modes (ARTSOM) algorithm is proposed to improve train punctuality and give a series of discrete operation modes (full traction, cruising, coasting, full braking), and thus to produce a feasible training set for DDPG, which can speed up the training process. Numerical experiments show that an optimized speed profile can be generated by DDPG within seconds on a realistic railway line. In addition, the results demonstrate the generalization ability of trained DDPG in solving TTO problems with different running times and line conditions.**

***Index Terms*— High-speed railway, train trajectory optimization, deep deterministic policy gradient, energy efficiency.**

I. Introduction

HIGH-SPEED railways (HSRs) is an important form of public transport because it can provide frequent, safe, fast and convenient transport services to the majority of citizens [1]. Safe and efficient train operation is supported by the train operation control systems, which are regarded as the brain and central nervous systems of HSRs. In practice, high-speed train drivers rely on their own experience to run a train in the daily operation, supervised by an automatic train protection system. In Japan, France, Germany, China and other countries, HSRs have experienced great development, but they still use manually driven trains. With the development of communication, control and computer technologies, HSRs are moving towards more automation. In recent years, scientific researchers and practitioners have explored ways to put automatic train operation (ATO) in practice, which is already widely used in the metro urban metro system.

As an important function of the ATO system, a recommended train speed profile has to be generated before the train departs, which is regarded as the target of the train control. The process of generating an optimal speed profile is called the train trajectory optimization (TTO) problem. When disturbances occur, the speed profile needs to be re-optimized as soon as possible according to the rescheduled running times assigned by the dispatching system. In 1968, Ichikawa [2] first formulated the TTO problem as the problem of optimal control on level tracks in which the model was solved by Pontryagin's Maximum Principle (PMP). Taking into account the non-linear constraints during operation such as varying gradients and speed limits, the problem becomes very difficult to solve. A generic approach to optimal control problems can be found in [3] where the problem is transcribed into a nonlinear programming problem and solved by nonlinear programming (NLP) methods. Also, heuristic algorithms are applied to solve the TTO problem by an iterative procedure. A comprehensive overview of the formulations and solutions for the TTO problem can be found in [4].

The methods for the TTO problem based on PMP aim at solving the differential equations for the optimal control where the optimal sequence of regimes is determined with their switching points. Based on PMP, Howlett [5] formulated a generalized problem with time as the independent variable for continuous train control to find an optimal control strategy, and necessary conditions were derived to calculate the optimal switching points. Howlett *et al.* [6] proposed a method of local energy minimization to calculate the switching points on tracks with steep gradients. Albrecht *et al.* [7] proved that the optimal switching points of regimes in each steep section of the track are uniquely defined, which deduces the uniqueness of the global optimal strategy. An implementation of the algorithm was reported in *Energymiser* to provide driving advice for train drivers. Liu and Golovitcher [8] reformulated the train optimal control problem using the train position as the independent variable. Their algorithm consists of two loops in which the outer loop finds the cruising speed with constant line resistance

Manuscript received 17 December 2020; revised 7 June 2021; accepted 15 July 2021. Date of publication 25 August 2021; date of current version 9 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61925302 and Grant 61790573, in part by the State Key Laboratory of Rail Traffic Control and Safety under Contract RCS2020ZZ002, in part by Beijing Jiaotong University, and in part by the Key Project of China Railway Beijing Bureau Group Company, Ltd., under Grant 2020AY03. The Associate Editor for this article was F. Chu. *(Corresponding authors: Hairong Dong; Min Zhou.)*

Lingbin Ning, Min Zhou, Zhuopu Hou, and Hairong Dong are with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (e-mail: zhoumin@bjtu.edu.cn; hrdong@bjtu.edu.cn).

Rob M. P. Goverde is with the Department of Transport and Planning, Delft University of Technology, 2628 CN Delft, The Netherlands.

Fei-Yue Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Digital Object Identifier 10.1109/TITS.2021.3105380

on each interval to compute the required trip time, and the inner loop builds the optimal control strategy for the speed found in the outer loop.

To deal with the discontinuous operational constraints such as varying gradients and speed limits, the TTO problem can be transcribed in a NLP problem by discretizing the running distance to a sequence of intervals with respects to the speed limits, gradients, tunnels, and so on. With a piecewise affine approximation to the nonlinear terms of maximum traction force and energy consumption, Wang *et al.* [9] transcribed the train trajectory optimization problem as a mixed-integer linear programming (MILP) problem. The TTO problem can be formulated as a multiple-phase optimization model. Wang and Goverde [10] solved the TTO problem by a pseudospectral method, where the signaling constraints were taken into consideration for the safe operation of two successive trains. Wang and Goverde [11] extended their model to a multi-train trajectory optimization model considering deviations from a timetable with the objectives of energy savings and delay recovery.

In order to address the problems of manual driving, multi-train operations or other complex operational scenarios, heuristic algorithms can be used to solve the TTO problem. Generally, these algorithms optimize the train speed profile based on the knowledge of the optimal control sequences derived from PMP. Sicre *et al.* [12] proposed a genetic algorithm to solve the TTO problem. Fuzzy parameters embodied in the algorithm were involved to represent the uncertainty of manual driving. Liu *et al.* [13] developed a cooperative model to formulate the multiple trains' operation on the same electrical section in which a heuristic algorithm was proposed to find out the switching points for maximizing the utilization of regenerative energy. A mixed structure combined with offline and online optimization methods was established by Li *et al.* [14] to achieve an energy-saving and high punctuality trajectory. A genetic algorithm (GA) was applied to obtain an offline optimized train trajectory that is used to calculate a new energy-saving profile for the remaining route in the online optimization to ensure punctuality. In order to assure an approach to railway stations, Allotta *et al.* [15] proposed an ATO design based on discrete position feedback. Further, Pugi *et al.* [16] proposed a gain-scheduled transfer function to compute smooth speed profiles. To model the uncertainty in manual driving utilizing fuzzy parameters, Fernández-Rodríguez *et al.* [17] proposed an approach based on the NSGA-III algorithm to solve the eco-driving problem in real time. An objective, the risk of delay in arrival, was defined to evaluate the time margin of the train up to the destination in the train operation process.

With the widespread application of artificial intelligence (AI) algorithms, especially deep learning and reinforcement learning, also AI algorithms were developed to solve the problems of a railway system. For the train timetable rescheduling problems, Šemrov *et al.* [18] presented a Q-learning method to reschedule trains on a single track railway after a disturbance. Another Q-learning method [19] was proposed by Jiang *et al.* to solve the problem of stranded passengers on the platforms of urban rail transit systems. Ning *et al.* [20] developed an approach based on deep reinforcement learning to adjust the running time and orders of trains under disturbance, which aims to minimize the delays of all trains at all stations. Under the same circumstance, another method based on the Monte Carlo Tree Search was developed by Wang *et al.* [21] to solve the problem. For urban metro trains in solving the TTO problem, Yin *et al.* [22] proposed two intelligent train operation algorithms based on an expert system and reinforcement learning (RL). However, for high-speed railways, there is limited work about AI algorithms applied to TTO problems.

In addition, taking into account the timetable rescheduling problem, the integration of timetable rescheduling and train operation is a promising future direction [23]–[25]. Ning *et al.* [26] proposed a basic framework for the integration of timetable rescheduling and train operation for high-speed railways, which considered information of both aspects at the same time. The integrated framework involved a wide range of information exchanged between the timetable rescheduling layer and the train operation layer, which puts forward higher requirements for the optimization of the train speed profile.

A novel approach based on DDPG is developed to overcome difficulties of large solution space, and meet requirements of generating high-quality solutions in real time under complex constraints, which involve varying speed limits, gradients, and scheduled/rescheduled various running times. In the structure of the proposed approach, deep learning is used to capture the features of train operation states, including running time, speed limits and gradients, while the deterministic policy gradient algorithm is used to generate the actions to get a sequence of speed value. The DDPG algorithm performs a great deal of offline training and stores the knowledge learned from the training process into neural networks, to be able to generate a series of train trajectories online. In summary, the contributions are as follows:

1) A novel learning-based approach DDPG is proposed to obtain an energy-saving train trajectory based on an offline training from the interaction between the agent and the trajectory simulation environment.
2) The DDPG is fast enough to generate an optimized train trajectory in real time. Taking the advantages of deep learning in environment perception and nonlinear mapping, the DDPG could learn the nonlinear relation between the constraints and actions, which is stored in the structure of DL and restored for online utilization.
3) The ARTSOM algorithm is proposed that heuristically allocates the available running time over the successive segments and selects a sequence of operation modes to provide high-quality samples for DDPG in the offline training to accelerate the convergence of the agent.

The remainder of this paper is organized as follows. In Section II, the necessary parameters and variables are first defined for the problem statement and the driving performance. Then, the assumptions are given and the TTO problem is formulated and the ARTSOM algorithm is proposed. Section III illuminates basic principles and components of the DDPG approach to the TTO problem. Also, the indicators of driving performance are given in this section. Numerical experiments

TABLE I
PARAMETERS OF THE TTO PROBLEM

N, i	Number and index of segments, $i = 1, 2, ..., N$
J, j	Number and index of segment sets
m	Mass of train
x	Running position
$v(x)$	Running speed at position x
$A(v)$	Maximum traction effort at speed v
$B(v)$	Maximum braking effort at speed v
$F(v, x)$	Resultant effort of maximum traction or braking effort at speed v at position x
$D(v)$	Basic resistance of train at speed v
$G(x)$	Line resistance of train at position x
$t(x)$	Running time at position x
$\tilde{V}(x)$	Speed limits along the lines at position x
x_0, x_f	Starting point and ending point
T	Practical running time of the trajectory
E	Energy consumption of the trajectory
s_i	Segments divided by points of change in the speed limit $\tilde{V}(x)$
b_{i-1}, b_i	Boundaries of the segment s_i
l_i	Length of segment s_i
t_i	Allocated running time of segment s_i
$t_i^{\min}$	Minimum running time of segment s_i
T_p	Scheduled running time from timetable
$T_{\min}$	Minimum running time in practical operation
T_ts	Total running time supplement
$\bar{v}_i^\mathrm{s}$	Average speed of segment s_i
o_j	Segment set of same average speed
O	Ordered segment sets
$\bar{v}_j$	Speed of segment set o_j
$\bar{V}$	Ordered speeds of segment sets
L_j	Total length of segment set o_j
t_i^s	Running time supplement of segment s_i
T_rs	Required time supplement of segment set o_1
$\bar{v}_\mathrm{r}(x)$	Remaining average speed
$c(x)$	Additional speed of the segment at position x
$v_\mathrm{r}(x)$	Reference speed with real-time operation information
$M(x)$	Train operation mode selected by the proposed SOM algorithm
$p(x)$	Percentage of the maximum traction or braking effort at position x

are established and the performance of the proposed approach is analyzed in Section IV. Finally, conclusions are summarized in Section V.

II. PROBLEM DESCRIPTION AND FORMULATION

A. Parameters and Decision Variables

For a better understanding of the TTO problem, the parameters and decision variables are listed in TABLE I.

B. Assumptions

In order to simulate the train operation and generate a series of feasible train speed profiles to speed up the training process of DDPG, the assumptions are listed as follows:

1) The environment focuses on the simulation of a single train operation. Therefore, it is assumed that the end of Movement Authority (MA) for the train is set to the destination station.
2) The purpose of this article is to drive the train to the destination station with the minimum traction energy rather than the resultant energy considering regenerative braking energy. The inclusion of regenerative braking leads to braking from a higher speed by later coasting using more traction energy that is compensated by the regenerative braking [27]. Therefore, to implicitly discourage regenerative braking energy to compensate for unnecessary traction, it is assumed that regenerative braking energy is not considered.

C. Model Formulation

Generally speaking, a train can be simplified as a single-point mass and is subject to different forces during its run between stations. These forces consist of the traction effort, the braking effort, the basic resistances, and the additional resistances. In practice, traction effort and braking effort are not allowed to act on the train at the same time, so the motion of a train can be formulated as

$$\frac{dv(x)}{dx} = \frac{p(x) \cdot F(v, x) - D(v) - G(x)}{\rho \cdot m \cdot v(x)}, \tag{1}$$

where ρ is the rotating mass factor, $F(v, x)$ denotes the maximum traction effort $A(v)$ or braking effort $B(v)$, which should satisfy the following equation

$$F(v, x) = \kappa_a(x) \cdot A(v) + \kappa_b(x) \cdot B(v), \quad \kappa_a(x), \quad \kappa_b(x) \in \{0, 1\} \tag{2}$$

where $\kappa_a(x) = 1, \kappa_b(x) = 0$, if traction effort is applied, $\kappa_a(x) = 0, \kappa_b(x) = 1$, if braking effort is applied, and otherwise $\kappa_a(x) = \kappa_b(x) = 0$. The maximum traction effort $A(v)$ and braking effort $B(v)$ are specified by the curves of maximum electrical traction effort and electrical braking effort, which are functions of train speed [28]. $p(x)$ is the percentage of the maximum traction or braking effort at position x. $D(v)$ denotes the basic resistances including friction resistance and air resistance, which are described as a quadratic function of speed according to the Davis formula. $G(x)$ represents the additional resistances involving the gradient resistance, the curve resistance, and the tunnel resistance at train position x [29]. The elapsed time $t(x)$ satisfies the differential equation

$$\dot{t}(x) = \frac{dt(x)}{dx} = \frac{1}{v(x)}. \tag{3}$$

To guarantee safety, the train speed is not allowed to exceed the speed limits along the line, so the train speed should satisfy the speed limit constraint

$$v(x) \le \tilde{V}(x). \tag{4}$$

The train starts from the starting point and stops at the ending point, so the speed should satisfy the constraints

$$v(x_0) = v(x_\mathrm{f}) = 0. \tag{5}$$

D. ARTSOM Algorithm

To guarantee the punctuality of train operation, a heuristic algorithm ARTSOM is proposed, which involves two processes: allocating running time and selecting operation modes.

1) Allocating Running Time (ART): The punctuality of train operations is one of the most critical factors of the high-speed railway system [30]. To achieve the goal of high punctuality, the driver or ATO controls the train to operate on time according to a timetable. A scheduled/rescheduled running time may be achieved by countless feasible trajectories, but there is only one with the minimum energy consumption [7].

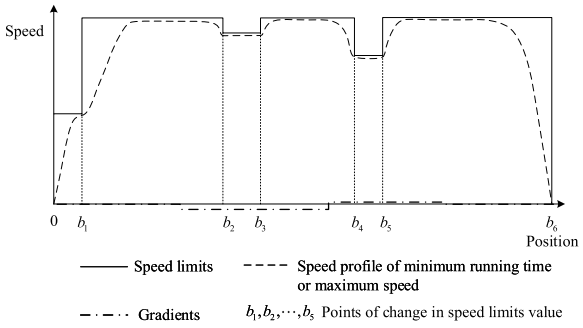


Fig. 1. Minimum running time speed profile.

The time difference between the assigned running time and the minimum running time, that is, the running time supplement, should be distributed to different segments to reduce the operating energy consumption. Moreover, according to the trajectory of the minimum running time, the segments with higher speed should have a higher priority to obtain additional running time, because of the fact that the energy used to accelerate the train is proportional to the square of the speed. An algorithm is proposed to sequentially allocate time supplement to those segments with the highest speed until all the time supplement has been made.

First, by dividing the whole section between the adjacent stations into several segments with respect to points of change in speed limits value $\tilde{V}(x)$, the boundaries of segments are defined as $[b_0, b_1, \ldots, b_i, \ldots, b_N]$ and the lengths of segments as $[l_1, l_2, \ldots, l_i, \ldots, l_N]$. To get the running times of segments expressed by $[t_1, t_2, \ldots, t_i, \ldots, t_N]$, the trajectory of the minimum running time needs to be calculated, corresponding to running as fast as possible while respecting the supervised speed profile from the automatic train protection (ATP) system, as shown in Fig. 1. So, the minimum running times of segments can be described as $[t_1^{\min}, t_2^{\min}, \ldots, t_i^{\min}, \ldots, t_N^{\min}]$, which are obtained from the trajectory of the minimum running time and initialized to the allocated running time of segments in the initial calculation step, that is, $[t_1, t_2, \ldots, t_i, \ldots, t_N] = [t_1^{\min}, t_2^{\min}, \ldots, t_i^{\min}, \ldots, t_N^{\min}]$. The total running time supplement can be calculated by

$$T_{\min} = \sum_{i=1}^{N} t_i^{\min}, \tag{6}$$

$$T_{\text{ts}} = T_{\text{p}} - T_{\min}, \tag{7}$$

where T_{p} and $T_{\min}$ are the scheduled running time from the timetable and minimum running time in practical operation, respectively. According to the length and the running time of the segment, the average speed of segment s_i can be given as

$$\bar{v}_i^{\text{s}} = \frac{l_i}{t_i}, \quad i = 1, 2, \ldots, N. \tag{8}$$

To allocate the total running time supplement T_{ts} to some specific segments, the set of segments is divided into J different segment sets o_j based on the average speed. Note that it is possible that o_j contains one or more, not necessarily adjacent, segments s_i. The segment sets o_j are sorted according to their

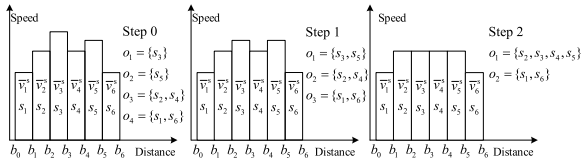


Fig. 2. Process of allocating the total running time supplement to segments.

average speeds to get the ordered segment sets

$$\bar{O} = [o_1, \ldots, o_J], \tag{9}$$

and associated ordered speeds of segment sets given as

$$\bar{V} = [\bar{v}_1, \ldots, \bar{v}_J], \tag{10}$$

where $\bar{v}_1$ and $\bar{v}_J$ are the biggest and smallest average speeds. So, the total length of each segment set o_j can be calculated by

$$L_j = \sum_{i:s_i \in o_j} l_i. \tag{11}$$

Then, $\bar{v}_1$ and $\bar{v}_2$ are selected, i.e., the biggest and the second biggest untraversed elements in the ordered speeds of segment sets $\bar{V}$, to calculate the required time supplement T_{rs} for segment set o_1 by

$$\Delta T_{\text{rs}} = \frac{1}{\bar{v}_2} - \frac{1}{\bar{v}_1}, \tag{12}$$

$$T_{\text{rs}} = \Delta T_{\text{rs}} L_1. \tag{13}$$

If the required time supplement T_{rs} is not greater than the total running time supplement T_{ts}, we define the running time supplement t_i^{s} of segment s_i

$$t_i^{\text{s}} = \frac{l_i}{L_1} T_{\text{rs}}, \quad \forall i : s_i \in o_1. \tag{14}$$

The allocated running time t_i of segment s_i is then updated by

$$t_i = t_i^{\text{s}} + t_i^{\text{old}}, \quad \forall i : s_i \in o_1. \tag{15}$$

where t_i^{old} is the last value of the allocated running time. The new average speed of segment s_i can be calculated as

$$\bar{v}_i^{\text{s}} = \frac{l_i}{t_i}, \quad \forall i : s_i \in o_1. \tag{16}$$

Then the total running time supplement for the next step is updated to

$$T_{\text{ts}} = T_{\text{ts}}^{\text{old}} - T_{\text{rs}}, \tag{17}$$

and the ordered sets $\bar{O}$ and $\bar{V}$ are updated according to (9) and (10). The segments $s_i \in o_1$ are allocated a proper period of running time, the speed $\bar{v}_1$ of segment set o_1 reduces to $\bar{v}_2$, so the segment set o_2 can merge into the segment set o_1 and the set o_2 is replaced by the set o_3. This process repeats until all the running time supplement T_{ts} is allocated to the relevant segments, as illustrated in Fig. 2.

If $T_{\text{rs}} > T_{\text{ts}}$, then the remaining running time supplement is allocated to segment set o_1

$$t_i^{\text{s}} = \frac{l_i}{L_1} T_{\text{ts}}, \quad \forall i : s_i \in o_1. \tag{18}$$

Note that the speeds $\bar{v}_i^s$ for $s_i \in o_1$ reduce to $\bar{v}_2$, and thus o_2 is expanded with o_1. This can be proved as follows. For all i with $s_i \in o_1$ we can rewrite (16) by substituting (12), (13), (14), and (15) as

$$\begin{aligned}\bar{v}_i^{\mathrm{s}} &= \frac{l_i}{t_i^{\mathrm{old}} + \frac{l_i}{\bar{v}_2} - \frac{l_i}{\bar{v}_1}} \\ &= \frac{1}{\frac{t_i^{\mathrm{old}}}{l_i} + \frac{1}{\bar{v}_2} - \frac{1}{\bar{v}_1}} \\ &= \frac{1}{\frac{1}{\bar{v}_1} + \frac{1}{\bar{v}_2} - \frac{1}{\bar{v}_1}} \\ &= \bar{v}_2. \end{aligned} \tag{19}$$

This proves that the average speeds are equal to that of segments from o_2.

Finally, based on these definitions and derivations, the process of allocating running time is described in Algorithm 1. After obtaining specific the running time for each segment, the operation modes need to be specified for actually controlling the train.

2) Selecting Operation Mode (SOM): Based on the average speed of the segments, a heuristic algorithm is developed to determine train operation modes, which almost follow the optimal control sequences derived from PMP. Here, drawing on the idea of feedback from control theory, the framework of the SOM algorithm needs a reference speed, which is compared with the current speed to adjust the mode of the train. There are three parameters that divide the error threshold between the current speed and the reference speed into four areas, which correspond to the four operating modes of the train, that is, full traction, cruising, coasting and full braking. The mode of full traction should be selected when the current speed is lower than the reference speed, and otherwise, one of cruising, coasting or braking depending on the degree of the error. Another algorithm based on the DDPG is proposed to provide a continuous rate of acceleration or deceleration in the following section.

First, we define a reference speed which is composed of two parts

$$v_{\mathrm{r}}(x) = \bar{v}_{\mathrm{r}}(x) + c(x) \tag{20}$$

where $\bar{v}_{\mathrm{r}}(x)$ is the remaining average speed, given by

$$\bar{v}_{\mathrm{r}}(x) = \frac{x_{\mathrm{f}} - x}{T_{\mathrm{p}} - t(x)}. \tag{21}$$

and $c(x)$ is defined below. The remaining running time can be guaranteed if the SOM algorithm can control the train speed to track this target. It is reasonable that the remaining average speed is regarded as the tracking target of the whole remaining section when the scheduled running time is much larger than the minimum running time. However, when the running time supplement is small, that is, the train trajectory is very close to the trajectory of the minimum running time, the tracking target needs to reflect the detailed information of the segments. Therefore, we define $c(x)$ to represent the additional speed of the segment at position x, given by

$$c(x) = z(T_{\mathrm{p}} - T_{\min}) \cdot \bar{v}_i^{\mathrm{s}}, \quad i = s_{\mathrm{cur}}(x), \tag{22}$$

Algorithm 1 Allocating Running Time (ART)

Input: the scheduled running time from timetable T_{p}; speed limits $\tilde{V}$ along the line ;

Output: the average speed of segments $[\bar{v}_1^{\mathrm{s}}, \bar{v}_2^{\mathrm{s}}, \ldots, \bar{v}_i^{\mathrm{s}}, \ldots, \bar{v}_N^{\mathrm{s}}]$;

```
Divide the operation distance into N segments [s_1, s_2, ..., s_i, ..., s_N] according to the speed limits Ṽ;
Compute the trajectory of minimum running time and obtain the minimum running times t_i = t_i^min and T_min by (6);
Calculate the total running time supplement T_ts by (7);
Calculate the average speeds of segments v̄_i^s by (8);
Calculate the ordered segment sets Ō by (9), the ordered speeds of segment sets V̄ by (10) and the total length L_j of each set o_j ∈ Ō by (11);
Calculate the required time supplement T_rs for segment set o_1 by (12) and (13);
while T_rs ≤ T_ts do
    for all segments s_i in set o_1 do
        Calculate the average time t_i^s for segment s_i by (14);
        Calculate the allocated time t_i for segment s_i by (15);
        Calculate the allocated segment average speed v̄_i^s for segment s_i by (16);
    Update the total running time supplement T_ts by (17);
    Update the ordered segment sets Ō by (9), the ordered speeds of segment sets V̄ by (10) and the total length L_j of each set o_j ∈ Ō by (11);
    Update the required time supplement T_rs for segment set o_1 by (12) and (13);
for all segments s_i in segment set o_1 do
    Calculate the running time supplement t_i^s for segment s_i by (18);
    Calculate the allocated time t_i for segment s_i by (15);
    Calculate the allocated segment average speed v̄_i^s for segment s_i by (16);
Output the average speed of segments v̄_i^s.
```

where $s_{\mathrm{cur}}(x)$ denotes the index of the segment in which the train is running at position x. $z(x)$ is a strictly decreasing function [18], given by

$$z(x) = \frac{0.1}{1 + e^{\frac{10 \cdot (x - 0.4 \cdot d_{\max})}{d_{\max}}}} + 0.05, \tag{23}$$

where $d_{\max}$ is the maximum time supplement. It is assumed that the maximum rescheduled running time is not more than twice the minimum running time. Based on the line data in the following section, $d_{\max}$ is set to 700s. $z(x)$ decreases the proportion of the average speed as the running time supplement increases.

According to the difference between practical running speed $v(x)$ and reference speed $v_{\mathrm{r}}(x)$, the SOM algorithm at distance

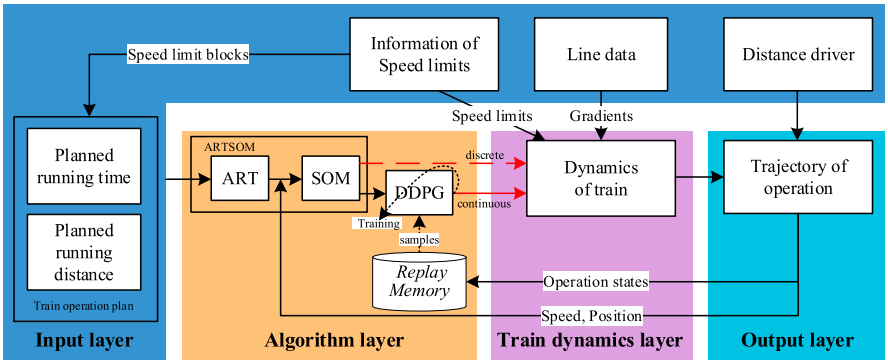


Fig. 3. Simulation framework for TTO problem with DDPG algorithm.

x can be described by

$$M(x) = \begin{cases} \text{Tr} & 0 \le v(x) < v_{\text{r}}(x) \cdot C_{\text{tr}} \\ \text{Cr} & v_{\text{r}}(x) \cdot C_{\text{tr}} \le v(x) < v_{\text{r}}(x) \cdot C_{\text{cr}} \\ \text{Co} & v_{\text{r}}(x) \cdot C_{\text{cr}} \le v(x) < v_{\text{r}}(x) \cdot C_{\text{co}} \\ \text{Br} & \text{otherwise}, \end{cases} \tag{24}$$

where $\text{Tr}, \text{Cr}, \text{Co}$ and Br represent full traction, cruising, coasting, full braking modes, respectively. C_{tr}, C_{cr} and C_{co} are threshold coefficients related to the train operation mode which are able to dynamically adjust the range for selecting each mode. The parameter values $C_{\text{tr}} = 0.93$, $C_{\text{cr}} = 1.16$ and $C_{\text{co}} = 1.5$ are obtained by the experiments of sensitivity analysis, in which 16 case studies are conducted to demonstrate the performance of DDPG (as shown in Fig. 17). In particular, when the current speed of the train is near the speed limit, it will stop traction and start cruising, coasting or braking depending on the ATP curve. Note that the output of SOM is a discrete operation mode which means that the percentage $p(x)$ of the maximum traction or braking effort at position x is set to 100%. In the following section, the DDPG algorithm is introduced to adjust the percentage $p(x)$, which is applied to produce a series of continuous actions for optimizing train speed profiles. The distance between the successive calculation points is set to 30m. Then the dividing points of the segments are inserted into the sequence of calculation points in order. The heuristic mode selection function $M(x)$ is defined for a reference speed that is based on the remaining average speed but it does not take into account intermediate speed restrictions. Therefore, the selected mode may have to be corrected to follow the ATP speed supervision curve in case it would exceed this curve. Based on these definitions and derivations, the selection process of the train operation mode is described in Algorithm 2.

As shown in Fig. 3, a comprehensive flow chart has been established to illustrate the structure of the proposed approach, which consists of four layers, i.e., the Input layer, the Algorithm layer, Train dynamics layer, and the Output layer. According to the information from the Input layer, including the line data, speed limits, running time and distance, ART allocates the running time to the successive segments divided by speed limit changes and SOM gives the discrete train operation modes. Based on these discrete modes, the DDPG algorithm generates continuous actions for accurate train speed control. Then, the control strategy derives the train movement on the line under its dynamics. The Output layer depicts the whole optimized speed profile and evaluates its indicators. In the loop containing the red dashed arrow, SOM selects discrete modes to derive the train movement and stores the operation data in the *Replay Memory* in the phase of preparing the training set. After that, a certain amount of training episodes are performed for preliminary training of DDPG. Then, DDPG generates continuous actions based on the discrete modes from SOM to control the train, where the operation data gradually updates the *Replay Memory*, as shown in the loop containing the red solid arrow.

Algorithm 2 Selecting Operation Mode (SOM)

Input: the segment running times $[t_1, t_2, \ldots, t_i, \ldots, t_N]$;
Output: the practical train mode $M(x)$;
Initialize the threshold coefficients C_{tr}, C_{cr} and C_{co};
Input the current position x;
Locate the segment $s_{\text{cur}}(x)$;
Calculate the reference speed $v_r(x)$ by (20);
Select the operation mode $M(x)$ by (24);
If the operation mode M(x) exceeds the ATP curve then follow the ATP curve instead.

III. Principles of Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient (DDPG), a recently developed approach in reinforcement learning by Google Deepmind [31], is a policy learning method that integrates deep learning neural networks into Deterministic Policy Gradient (DPG) [32]. Different from the basic idea of value-based reinforcement learning algorithms such as DQN [33] and its improved versions i.e., Nature DQN [34], Prioritized Replay DQN [35], Double DQN [36], or Dueling DQN [37], that calculate the values of actions at the states and then greedily choose an action according to its value, the policy-based method chooses the action directly according to the current state, which omits the intermediate step of evaluating each action value.

DDPG is one of the advanced algorithms in deep reinforcement learning (DRL), which combines the advantages of deep learning (DL) and deterministic policy gradients (DPG) algorithm with the Actor-Critic structure. The advantages of DDPG

are summarized as follows: 1) deep neural networks are used as strategy and value functions, which can be trained based on the Actor-Critic structure. 2) DDPG will be able to deal with high-dimensional discrete or continuous action sets. 3) the neural networks can be updated at each timestep. Following a standard reinforcement learning setup, the structure contains three basic parts: *Environment* and *State*, *Agent* and *Action*, and *Reward*. In the following subsection, the details of these three parts are presented to solve the TTO problem.

A. Environment and State

To solve the TTO problem, an environment is established based on the model described in Section II which includes the train dynamics, speed limits, gradients, and the boundary conditions. After executing an action, the environment computes the acceleration, speed, and spent time of the train. Meanwhile, the driving performance indicators of punctuality and energy consumption are given by this environment. In this paper, the train operation states involves speed limits, gradients, running distance, running speeds and running times, which are regarded as the inputs of deep learning to extract the features.

B. Agent and Action

In the real world, the train driver is responsible for safe, punctual, and efficient train operation. According to the scheduled running time from a timetable and the practical operating environment information (including train speed, position, speed limits, and gradients), the driver controls the train based on the knowledge learned from the previous driving experience. The learning agent in the reinforcement learning structure corresponds to the train driver, who is responsible for optimal train operations based on this information. To achieve accurate speeds in the cruising mode, the agent of DDPG is designed to generate continuous actions for a precise control strategy.

C. Reward

There is no explicit label for the samples during the interaction between the agent of RL and the environment, but an instant reward, and the expected return is the sum of the discounted rewards to guide the network training. In the TTO problem, because a train running state cannot fully reflect the difference between any trajectory and the optimal one, the reward is directly related to the train performance at the scheduled destination instead of any intermediate point. The objectives of the TTO problem involve punctuality and energy efficiency, so the two indicators of train driving performance are regarded as these factors of the reward function for the actions. The practical running time of the trajectory can be rewritten from (3)

$$T = \int_{x_0}^{x_\mathrm{f}} \frac{dx}{v(x)}. \tag{25}$$

The energy consumption is computed as the integral of traction effort over train running distance

$$E = \int_{x_0}^{x_\mathrm{f}} p(x) \cdot \kappa_a(x) \cdot A(v) dx. \tag{26}$$

The reward function can be defined as the function of the two indicators

$$r_t = \begin{cases} -\mathrm{C_m} & \exists x \in (x_0, x_\mathrm{f}), v(x) < 0 \\ -\dfrac{|T - T_\mathrm{p}|}{T_\mathrm{p}} - a\dfrac{E - E_\mathrm{ARTSOM}}{E_\mathrm{ARTSOM}} & \text{otherwise,} \end{cases} \tag{27}$$

where $\mathrm{C_m}$ $(= 5)$ is a relative big positive value, which is used to distinguish an infeasible solution if $\exists x \in (x_0, x_\mathrm{f}), v(x) < 0$. T_p is the scheduled running time, and E_ARTSOM is the energy consumption of a complete train trajectory from ARTSOM, depending on the given running time. a is a positive weight to balance the two objectives.

IV. Case Studies

This section demonstrates the adaptation of the train trajectory simulation environment, and the performance of ARTSOM and DDPG. The experiments are tested on an adapted section from Beijing-Shanghai high-speed railway, which is a line about 46km long with varying speed limits and gradients. The line data of gradients and speed limits are shown in TABLE II. The characteristics of the train in traction mode are provided by [28] based on practical experiments performed on a flat and straight track. According to the line data and the characteristics of the train, the calculation result of the minimum running time is 750s. The scheduled running time is set to 900s. The train trajectory simulation environment and proposed algorithms are implemented in Python 3.6.5 and use a deep learning framework, TensorFlow. The experiments are conducted on a Windows 10 X64 Professional Edition computer with a 2.5 GHz Intel Core i7 CPU and 12 GB RAM.

A. Basic Experiments

In this experiment, several scenarios with different running times are tested to demonstrate the effectiveness of ARTSOM, in which the trains running times are set to 760s, 900s, 1100s, and 1300s, respectively. To deal with the varying rescheduled running times, ART first allocates the total running time supplement to segments according to average speeds, as presented in Algorithm 1. Fig. 4 and Fig. 5 illustrate that the running time supplement is allocated to the segments by ART under different given running times, which indicates that the amount of time supplement increases and the average speed of segments decreases, as the given running time increases. SOM then selects the proper operation modes to control the train, as presented in Algorithm 2.

The speed profiles by ARTSOM with running times of 760s, 900s, 1100s and 1300s are shown in Fig. 6. The control regimes involve maximum traction effort, cruising at maximum speeds, coasting to the interact with ATP curve, and maximum braking effort to the end, which almost follow the optimal control sequences derived from PMP. Taking the given running times equal to 760s as an example, Fig. 7 shows the modes selected by SOM over distance and the areas of the four modes along with the reference speed. Here, the purple line illustrates the four modes with the numbers 0, 50, 100 and 150. In Fig. 7, before the speed limit, the coasting

TABLE II
SPEED LIMITS AND GRADIENTS DATA

Item	Value	Segment (m)
Speed Limits (km/h)	80	[0, 1000]
	300	(1000, 18000]
	250	(18000, 19500]
	300	(19500, 33000]
	280	(33000, 46110]
Gradients (‰)	0	[0, 1520]
	3	(1520, 6200]
	-8	(6200, 12310]
	3	(12310, 19090]
	10	(19090, 35100]
	0	(35100, 46110]

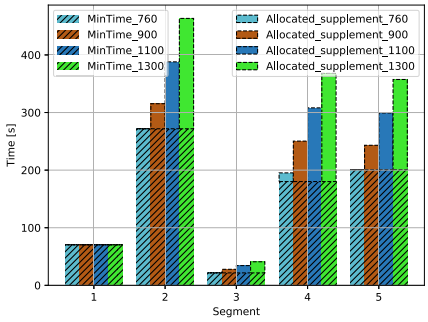


Fig. 4. Allocation of time supplement with running times 760s, 900s, 1100s, 1300s.

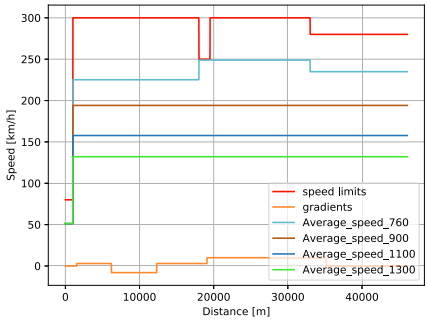


Fig. 5. Average speed of segments with running times 760s, 900s, 1100s, 1300s.

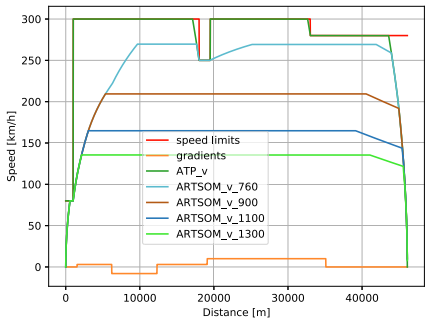


Fig. 6. Train trajectories from ARTSOM with running times 760s, 900s, 1100s, 1300s.

mode is not applied but cruising at some optimal cruising speed to obtain a suboptimal trajectory, which indicates that SOM will not select coasting at intermediate speed restrictions. Moreover, just before and during the speed restriction, the

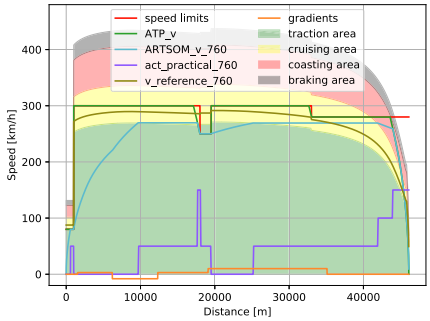


Fig. 7. Modes selected by SOM with running time 760s.

curve of ARTSOM will deviate from the reference speed and stay below the ATP curve and restricted speed until the end of the speed restriction, where the traction mode is used again to get back to the reference speed. In addition, the actual running times are almost equal to the corresponding scheduled ones, of which the maximum time error is 0.2% as shown in TABLE IV, indicating that the punctuality of train operation can be guaranteed.

In order to verify the efficiency of the proposed algorithm, we compare the ARTSOM with the genetic algorithm (GA), which is commonly adopted to solve the TTO problem [38], [39]. In the comparative experiment, each individual of the population is designed to involve five switching points to support two traction phases due to the restricted speed area halfway, which are necessary in the case where the given running time is close to the minimum one. 200 individuals are included in each generation and the algorithm runs for 100 iterations. The comparison of the train trajectory obtained by the methods of the GA and ARTSOM is shown in Fig. 8, In terms of the running time and the energy consumption, there is almost no difference between the speed profiles from GA and ARTSOM, while the computational time of the ARTSOM (about 1s) is two orders of magnitude smaller than that of the GA (about 200s for 25 iterations using Geatpy in Python). Fig. 9 shows the convergence of GA and indicates that the solution with the minimum objective can be found after 25 iterations. It can be concluded that compared with the GA algorithm, the ARTSOM algorithm can generate a speed profile that is almost equivalent to the energy consumption of the speed profile by GA in an almost real-time manner. The detailed comparison data under different running times are given in TABLE IV.

In addition, ARTSOM outputs discrete actions to generate complete train trajectories, as shown by the dashed red arrow in Fig. 3. These trajectories are placed in the *Replay Memory* for training the agent of DDPG. The well-prepared *Replay Memory* can provide a training set to speed up the convergence process of the neural network. Therefore, ARTSOM plays a critical part in the preparation of initial *Replay Memory* and the early training stage of DDPG. The parameters of the training process and DDPG structure are given in TABLE III.

B. TTO With Scheduled Running Time

In this experiment, we study the utility of the DDPG for the TTO problem with the scheduled running time. Based

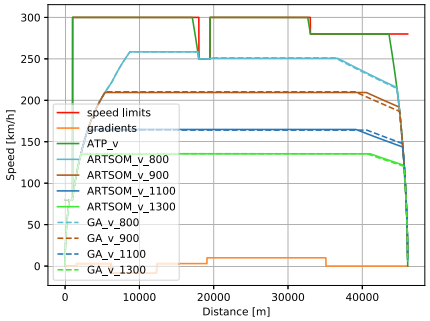


Fig. 8. Train trajectories from GA and ARTSOM with running time 800s, 900s, 1100s, 1300s.

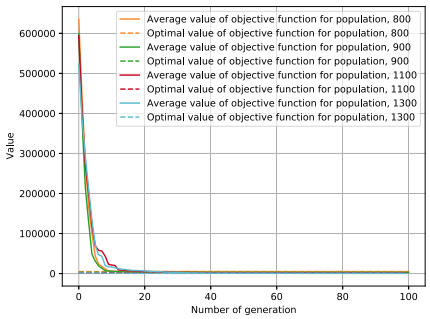


Fig. 9. Convergence of GA solving the TTO problem.

TABLE III
PARAMETERS IN THE PERFORMANCE PROCESS

Item	Value
Max episode number Ep	30000
Training episode e	[0, Ep]
Learning rate for current actor network α_a	0.001
Learning rate for current critic network α_c	0.002
Reward discount factor γ	0.9
Soft replacement factor τ	0.01
Replay memory size $\mathcal{M}$	10^8
Batch size B	128
Training time $T_{training}$	20 hours
Structure of actor network	30×20×2
Structure of critic network	30×20×1

on the discrete operation modes selected by ARTSOM, the DDPG agent is responsible for generating continuous actions to achieve any feasible acceleration for trains, especially when a train runs on a track with varying gradients in the cruising mode. The actor's output corresponds to a percentage of the maximum effort $F(v, x)$, as shown by the solid red arrow in Fig. 3.

At the stage of preparing training data, the DDPG only generates enough data for further expanding the *Replay Memory*, and then randomly samples batch data from the *Replay Memory* to train the neural networks and update them with newly generated data. Fig. 10 shows that the loss curve presents a downward trend with the training episode increasing. Due to the target networks being replaced by current networks, there are some fluctuating stages in the curve during the training process, in which the DDPG jumps out of the local optimal solution. As shown in Fig. 11, during the episodes from 0 to

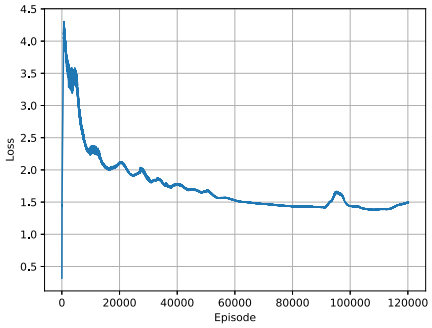


Fig. 10. The loss curve in the DDPG training process.

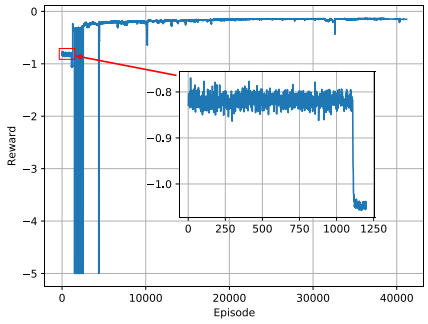


Fig. 11. The reward curve in DDPG training process.

about 1100, the beginning part of the loss curve is the stage of data preparation for *Replay Memory* in which ARTSOM is used to generate the feasible solutions with action randomness $\mathcal{N}$ of Gaussian distribution (mean $m_e = 0$ and decaying variance $v_e = 0.9995^e$). Then, during the episodes from 1300 to about 4500, the DDPG experiences the first training stage where the reward curve oscillates between infeasible values (=5) and feasible ones. In this stage, the networks of DDPG are trained from the stage that may generate infeasible solutions to the stage where all are feasible solutions. In the last stage of episodes from 4500 to training ends, the values of the reward curve rise steadily, which means that DDPG continues to produce feasible, and more and more valuable actions, as the training progresses. Note that in the process of generating each episode training data, an evaluation of actions and three pieces of training of DDPG are performed, so the episode number of loss is almost three times that of reward, except that no training is done during the data preparation stage.

Fig. 12 shows a comparison of the train trajectory obtained by the methods of the DDPG and ARTSOM. In the maximum traction phase, both methods have the same actions, so the two profiles coincide and their indicators are equal as shown in Fig. 13. On the one hand, DDPG uses the descending gradient to increase the train speed in the phase from 6200m to 6520m, which saves time. On the other hand, ARTSOM takes less time than the given running time, resulting in arriving 14s earlier. Both parts of the time are used by DDPG to run at a slower speed in the phase from 17000m to 45000m, while still arriving punctually. Although DDPG spends a little more energy than ART in the former phase, it saves much more

TABLE IV

PRACTICAL RUNNING TIME AND ENERGY CONSUMPTION OF GA, ARTSOM AND DDPG UNDER DIFFERENT SCHEDULED RUNNING TIMES

Scheduled/ Rescheduled time (s)	GA		ARTSOM				DDPG				
	Time (s)	Energy (kWh)	Time (s)	Time error (%)	Energy (kWh)	Energy saving P_G (%)	Time (s)	Time error (%)	Energy (kWh)	Energy saving P_G (%)	Energy saving P_A (%)
800	798	852.69	799	0.13	851.75	0.11	798	0.25	849.64	0.36	0.25
900	900	678.15	899	0.11	672.50	0.83	897	0.33	666.96	1.65	0.82
1000	997	588.49	998	0.20	589.94	-0.25	998	0.20	573.42	2.56	2.80
1100	1098	531.18	1100	0	532.89	-0.32	1100	0.00	509.34	4.11	4.42
1200	1202	487.46	1200	0	487.43	0.01	1199	0.08	460.84	5.46	5.46
1300	1301	455.21	1298	0.15	456.66	-0.32	1298	0.15	422.58	7.17	7.46

Time error = (Scheduled time - Time of ARTSOM or DDPG) / Scheduled time.
Energy saving P_G = (Energy of ARTSOM or DDPG - Energy of GA) / Energy of GA.
Energy saving P_A = (Energy of DDPG - Energy of ARTSOM) / Energy of ARTSOM.

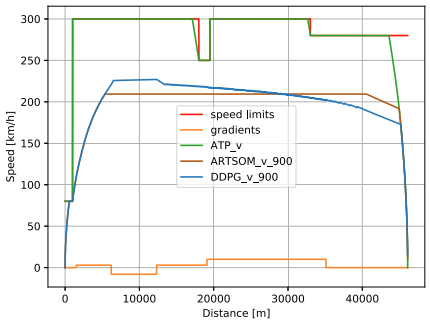


Fig. 12. Train trajectories from the DDPG and ARTSOM.

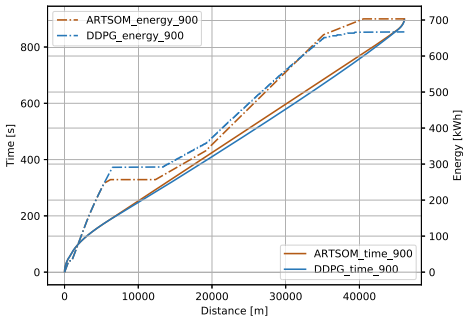


Fig. 13. Energy consumption and practical running time under the actions from DDPG and ARTSOM.

energy in the later phase. Fig. 13 depicts a comparison of two indicators of the train trajectory obtained with the methods of the DDPG and ARTSOM. The DDPG saves about 0.82% ($(672.50 - 666.96)/672.50 = 0.82\%$) energy over ARTSOM and running time is 899s, which means it can meet the requirement of punctuality.

C. TTO With Different Rescheduled Running Times

In this section, the effectiveness of DDPG for the trajectory optimization problem with rescheduled running times is illustrated with running times set to 800s, 1000s, 1100, 1200s, and 1300s. The trajectories of DDPG under different running times are described in Fig. 14. In the traction phase, all curves coincide while the ending speeds of the traction phase increase as the given running time decrease. From the overall view of Fig. 14, the less given running time, the greater the average speed, so the greater the energy consumption as shown in Fig. 15. Fig. 16 shows the running time under the control of DDPG which satisfies the requirement of the given running time from a timetable.

TABLE IV shows the running times and energy consumptions of train speed profiles by GA, ARTSOM and DDPG under different rescheduled running times. Although the energy consumption and running time of the speed profiles by GA and ARTSOM are not very different, ARTSOM could generate more energy-efficient speed profiles than that of GA in cases where the running times are less than 1000 seconds. However, when the running times are more than 1000 seconds, GA is able to find more energy-efficient speed profiles than ARTSOM. In terms of computational time, ARTSOM is significantly better than GA, because ARTSOM only needs to calculate a complete speed profile, while GA needs to calculate at least 5000 complete speed profiles in order to iterate 25 generations with 200 individuals per generation. Compared with GA and ARTSOM, DDPG can further improve energy efficiency to generate speed profiles with almost the same running times. As the running time changes from 800 seconds to 1300 seconds, the energy efficiency increases gradually. In addition, the energy-saving degree of DDPG in the cases with running times less than 1000s is significantly less than that in the cases with running times more than 1000s, indicating that there is less room for optimization of energy consumption in the former case, because more traction regime is used to accelerate the train so that the practical running times match the rescheduled ones.

Next, a sensitivity analysis is performed on the threshold coefficients in the selection of the operation mode algorithm. These coefficients C_{tr}, C_{cr} and C_{co} determine the range of selection of the four modes while the selection of the braking mode is constrained by the ATP profile. The 16 cases of the permutations of $C_{\text{tr}} \in \{0.89, 0.93, 0.97, 1.01\}$, $C_{\text{cr}} \in \{1.08, 1.12, 1.16, 1.20\}$ and $C_{\text{co}} = 1.5$ are used to demonstrate the performance of DDPG. Fig. 17 shows the energy-saving percentage of DDPG under the 16 cases of different threshold coefficients. It can be concluded that the permutation of ($C_{\text{tr}} = 0.93$, $C_{\text{cr}} = 1.16$) is a better one than the others.

In addition, we demonstrate the performance of DDPG under more complex line data with more speed limits and gradients. It can be seen from Fig. 18 that DDPG prefers to adopt the coasting mode to accelerate the train speed

TABLE V
PRACTICAL RUNNING TIME OF ARTSOM UNDER DIFFERENT GIVEN RUNNING TIME

Scheduled/ Rescheduled time (s)	ARTSOM			DDPG			
	Time (s)	Time error (%)	Energy (kWh)	Time (s)	Time error (%)	Energy (kWh)	Energy saving P_A (%)
1600	1601	-0.06	1557.30	1599	0.06	1546.68	0.68
1700	1702	-0.12	1342.88	1699	0.06	1327.61	1.14
1800	1799	0.06	1249.68	1801	-0.06	1178.27	5.71
1950	1950	0	1105.09	1951	-0.05	1037.41	6.12

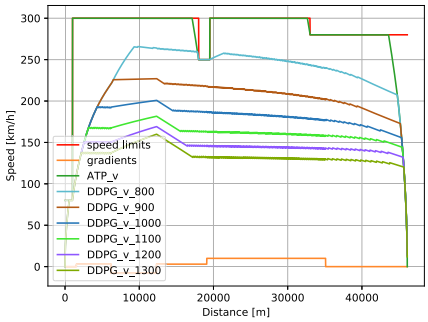


Fig. 14. Train trajectories from DDPG in the environment with different running time.

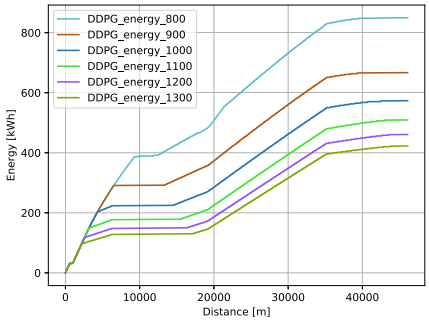


Fig. 15. The energy consumption the DDPG in the environment with different running times.

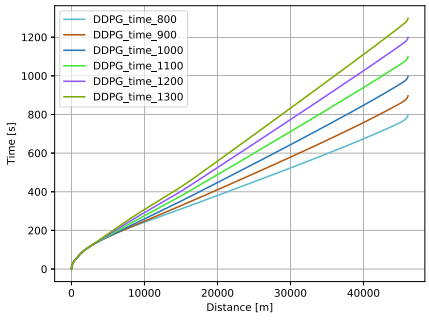


Fig. 16. The practical running time of the DDPG in the environment with different running times.

during the downhill section, which will save running time to compensate for the lower speed during other sections and result in energy saving. TABLE V gives the running times and energy consumption of ARTSOM and DDPG under different scheduled running times in detail, showing that DDPG has better energy utilization than ARTSOM.

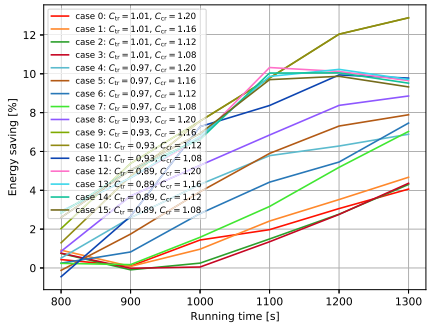


Fig. 17. Performance of DDPG under different threshold coefficients C_{tr}, C_{cr} and C_{co}.

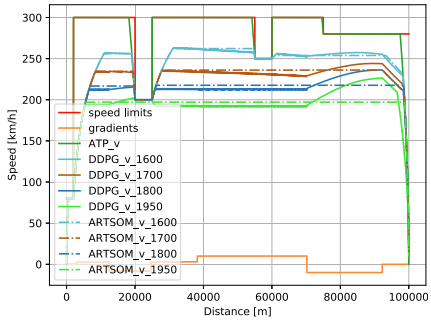


Fig. 18. Train trajectories from the DDPG and ARTSOM under an adapted line data with longer distance and more stringent speed limits and gradients.

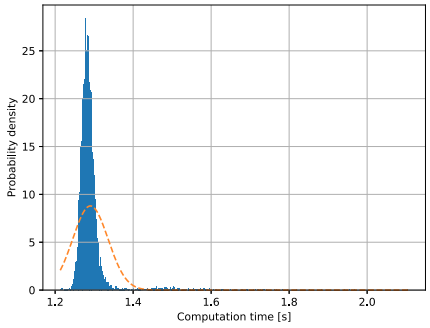


Fig. 19. Distribution of the computational times for 10000 experiments.

Furthermore, to verify that the DDPG can meet the real-time requirements in solving the TTO problem, 10000 experiments with the same running time based on the agent trained for 20 hours are carried out. As shown in Fig. 19, the computational time of 96.9% experiments to solve the problem is within 1.35s, and the maximum is 2.1s, It can be concluded that the proposed approach can achieve to generate optimized train speed profiles with energy saving in real time.

V. Conclusion

This paper proposes a learning-based approach for the high-speed train trajectory optimization problem which can generate an optimal recommended speed profile in real time. Based on the structure of Actor-Critic, an intelligent framework was designed for the train trajectory optimization problem in which extensive offline training is performed. The DDPG method learns the mapping relation between the observed states and actions by offline training, which is stored in the structure of deep learning and restored for online utilization. The proposed ARTSOM algorithm allocates the running time supplement to the segments with higher average speed and selects proper modes to control the train, which can produce feasible trajectories to build the training set and speed up the DDPG's training process.

The experiments were performed on an adapted section from the Beijing-Shanghai high-speed railway line, and results showed that the trained agent is able to generate punctual and energy-efficient train speed profiles with different scheduled/rescheduled running times. In addition, the computational times are within 3 seconds, which can meet the real-time requirement of solving the TTO problem. Compared with ARTSOM and GA, DDPG could reduce energy consumption by up to 7.46% and 7.17%, respectively. Based on the results of this paper, future research will improve the train trajectory simulator with more practical constraints and study the TTO problem with multiple stops.

References

[1] H. Dong, B. Ning, B. Cai, and Z. Hou, "Automatic train control system development and simulation for high-speed railways," *IEEE Circuits Syst. Mag.*, vol. 10, no. 2, pp. 6–18, May 2010.

[2] K. Ichikawa, "Application of optimization theory for bounded state variable problems to the operation of train," *Bull. Jpn. Soc. Mech. Eng.*, vol. 11, no. 47, pp. 857–865, 1968.

[3] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Philadelphia, PA, USA: SIAM, 2010.

[4] G. M. Scheepmaker, R. M. Goverde, and L. G. Kroon, "Review of energy-efficient train control and timetabling," *Eur. J. Oper. Res.*, vol. 257, no. 2, pp. 355–376, 2017.

[5] P. Howlett, "The optimal control of a train," *Ann. Oper. Res.*, vol. 98, nos. 1–4, pp. 65–87, 2000.

[6] 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.

[7] 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.

[8] R. R. Liu and I. M. Golovitcher, "Energy-efficient operation of rail vehicles," *Transp. Res. A, Policy Pract.*, vol. 37, no. 10, pp. 917–932, 2003.

[9] Y. Wang, B. De Schutter, T. J. van den Boom, and B. Ning, "Optimal trajectory planning for trains—A pseudospectral method and a mixed integer linear programming approach," *Transp. Res. C, Emerg. Technol.*, vol. 29, pp. 97–114, Apr. 2013.

[10] P. Wang and R. M. Goverde, "Multiple-phase train trajectory optimization with signalling and operational constraints," *Transp. Res. C, Emerg. Technol.*, vol. 69, pp. 255–275, Aug. 2016.

[11] P. Wang and R. M. Goverde, "Multi-train trajectory optimization for energy efficiency and delay recovery on single-track railway lines," *Transp. Res. B, Methodol.*, vol. 105, pp. 340–361, Nov. 2017.

[12] C. Sicre, A. Cucala, and A. Fernández-Cardador, "Real time regulation of efficient driving of high speed trains based on a genetic algorithm and a fuzzy model of manual driving," *Eng. Appl. Artif. Intell.*, vol. 29, pp. 79–92, Mar. 2014.

[13] J. Liu, H. Guo, and Y. Yu, "Research on the cooperative train control strategy to reduce energy consumption," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1134–1142, Sep. 2016.

[14] Z. Li, L. Chen, C. Roberts, and N. Zhao, "Dynamic trajectory optimization design for railway driver advisory system," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 1, pp. 121–132, Jan. 2018.

[15] B. Allotta, L. Chisci, P. D'Adamio, S. Papini, and L. Pugi, "Design of an automatic train operation (ATO) system based on CBTC for the management of driverless suburban railways," in *Proc. 12th Workshop Tech. Diagnostics, New Perspective Meas., Tools Techn. Ind. Appl. (IMEKO TC)*, 2013, pp. 84–89.

[16] L. Pugi, A. Reatti, F. Corti, and F. Grasso, "A simplified virtual driver for energy optimization of railway vehicles," in *Proc. IEEE Int. Conf. Environ. Electr. Eng., IEEE Ind. Commercial Power Syst. Eur. (EEEIC/I CPS Eur.)*, Jun. 2020, pp. 1–6.

[17] A. Fernández-Rodríguez, A. Fernández-Cardador, and A. P. Cucala, "Balancing energy consumption and risk of delay in high speed trains: A three-objective real-time eco-driving algorithm with fuzzy parameters," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 652–678, Oct. 2018.

[18] D. Šemrov, R. Marsetič, M. Žura, L. Todorovski, and A. Srdic, "Reinforcement learning approach for train rescheduling on a single-track railway," *Transp. Res. B, Methodol.*, vol. 86, pp. 250–267, Apr. 2016.

[19] Z. Jiang, J. Gu, W. Fan, W. Liu, and B. Zhu, "Q-learning approach to coordinated optimization of passenger inflow control with train skip-stopping on a urban rail transit line," *Comput. Ind. Eng.*, vol. 127, pp. 1131–1142, Jan. 2019.

[20] L. Ning, Y. Li, M. Zhou, H. Song, and H. Dong, "A deep reinforcement learning approach to high-speed train timetable rescheduling under disturbances," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3469–3474.

[21] R. Wang, M. Zhou, Y. Li, Q. Zhang, and H. Dong, "A timetable rescheduling approach for railway based on Monte Carlo tree search," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3738–3743.

[22] J. Yin, D. Chen, and L. Li, "Intelligent train operation algorithms for subway by expert system and reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2561–2571, Dec. 2014.

[23] X. Rao, M. Montigel, and U. Weidmann, "A new rail optimisation model by integration of traffic management and train automation," *Transp. Res. C, Emerg. Technol.*, vol. 71, pp. 382–405, Oct. 2016.

[24] L. S. Zhou, L. Tong, J. Chen, J. Tang, and X. Zhou, "Joint optimization of high-speed train timetables and speed profiles: A unified modeling approach using space-time-speed grid networks," *Transp. Res. B, Methodol.*, vol. 97, pp. 157–181, Mar. 2017.

[25] X. Luan, Y. Wang, B. De Schutter, L. Meng, G. Lodewijks, and F. Corman, "Integration of real-time traffic management and train control for rail networks—Part 1: Optimization problems and solution approaches," *Transp. Res. B, Methodol.*, vol. 115, pp. 41–71, Sep. 2018.

[26] B. Ning *et al.*, "Integration of train control and online rescheduling for high-speed railways: Challenges and future," *Acta Automatica Sinica*, vol. 45, no. 12, pp. 2208–2217, 2019.

[27] G. M. Scheepmaker and R. M. P. Goverde, "Energy-efficient train control using nonlinear bounded regenerative braking," *Transp. Res. C, Emerg. Technol.*, vol. 121, Dec. 2020, Art. no. 102852.

[28] *Characteristics of Train CRH380A in Traction Mode*. Accessed: May 8, 2015. [Online]. Available: https://wenku.baidu.com/view/6164a69755270722192ef7e7.html

[29] I. A. Hansen and J. Pachl, *Railway Timetable & Traffic*. Hamburg, Germany: Eurailpress, 2008.

[30] L. Meng and X. Zhou, "Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables," *Transp. Res. B, Methodol.*, vol. 67, no. 3, pp. 208–234, 2014.

[31] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*. [Online]. Available: http://arxiv.org/abs/1509.02971

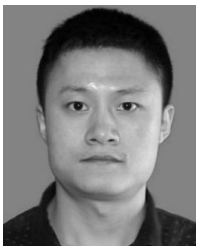
[32] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 387–395.

[33] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. [Online]. Available: http://arxiv.org/abs/1312.5602

[34] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[35] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015, *arXiv:1511.05952*. [Online]. Available: http://arxiv.org/abs/1511.05952

[36] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1–7.
[37] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," 2015, *arXiv:1511.06581*. [Online]. Available: http://arxiv.org/abs/1511.06581
[38] N. Zhao, C. Roberts, S. Hillmansen, and G. Nicholson, "A multiple train trajectory optimization to minimize energy consumption and delay," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2363–2372, Oct. 2015.
[39] S. Lu, S. Hillmansen, T. K. Ho, and C. Roberts, "Single-train trajectory optimization," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 743–750, Jun. 2013.

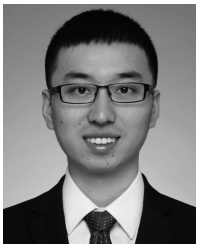


Lingbin Ning received the B.S. degree in automation from the North University of China, Taiyuan, China, in 2016. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University.

His current research interests include energy-efficient train operation, railway traffic management, and deep reinforcement learning.

Min Zhou (Member, IEEE) received the Ph.D. degree in traffic information engineering and control from Beijing Jiaotong University, Beijing, China, in 2019.

From November 2016 to November 2017, he was a Visiting Scholar with Ming Hsieh Department of Electrical Engineering, University of Southern California at Los Angeles, Los Angeles, CA, USA. He is currently a Postdoctoral Research Fellow with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University. His research interests include pedestrian dynamics, disturbance management and evacuation, fuzzy logic, and artificial intelligence.



Zhuopu Hou received the B.S. degree in automation from Tianjin Polytechnic University, Tianjin, China, in 2014. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University.

From November 2017 to November 2019, he was the Visiting Ph.D. student with the Center for Railway Research and Education, School of Electronic, Electrical and Computer Engineering, University of Birmingham, Birmingham, U.K. His current research interests include railway traffic management and optimization techniques.

Rob M.P. Goverde (Member, IEEE) received the M.Sc. degree in mathematics from Utrecht University, The Netherlands, in 1993, and the Professional Doctorate in Engineering (P.D.Eng.) degree in mathematical modeling and decision support and the Ph.D. degree in railway transportation from Delft University of Technology, The Netherlands, in 1996 and 2005, respectively. He is currently a Professor in railway traffic management and operations and the Director of the Digital Rail Traffic Laboratory, Delft University of Technology. He also chairs the Theme Railway Systems of the TU Delft Transport Institute. His research interests include railway timetabling, railway traffic management, disruption management, automatic train operation, and railway signaling. He cooperated in many international projects including the EU projects ON-TIME, MOVINGRAIL, RAILS, and PERFORMINGRAIL.

He is the Editor-in-Chief of the *Journal of Rail Transport Planning and Management*, an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, a Board Member of the International Association of Railway Operations Research (IAROR), and a fellow of the Institution of Railway Signal Engineers (IRSE).

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York, in 1990. He joined the University of Arizona in 1990 and became a Professor and the Director of the Robotics and Automation Laboratory (RAL) and Program in Advanced Research for Complex Systems (PARCS). In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Oversea Chinese Talents Program from the State Planning Council and "100 Talent Program" from CAS. In 2002, he was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory of Management and Control for Complex Systems. His current research focuses on methods and applications for parallel systems, social computing, and knowledge automation. He was the President of the IEEE ITS Society (2005–2007), Chinese Association for Science and Technology (CAST, USA) in 2005, the American Zhu Kezhen Education Foundation (2007–2008), and the Vice President of the ACM China Council (2010–2011). Since 2008, he has been the Vice President and Secretary General of Chinese Association of Automation. He is elected as a fellow of INCOSE, IFAC, ASME, and AAAS. He was the Founding Editor-in-Chief of the *International Journal of Intelligent Control and Systems* (1995–2000), the Founding EiC of *IEEE ITS Magazine* (2006–2007), the EiC of IEEE INTELLIGENT SYSTEMS (2009-2012), and the EiC of IEEE TRANSACTIONS ON ITS (2009–2016). Currently, he is the EiC of China's *Journal of Command and Control*. Since 1997, he has been serving as a General or the Program Chair of more than 20 IEEE, INFORMS, ACM, and ASME conferences.

Hairong Dong (Senior Member, IEEE) received the Ph.D. degree from Peking University in 2002. She is currently the Deputy Director of the National Engineering Research Center for Rail Transportation Operation Control System, and also a Professor with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China. Her research interests include intelligent transportation systems, automatic train operation, intelligent dispatching, and complex network applications. She was a Visiting Scholar with the University of Southampton in 2006 and the University of Hong Kong in 2008. She was also a Visiting Professor with the KTH Royal Institute of Technology in 2011.

She is currently a fellow of the Chinese Automation Congress and the Co-Chair of the Technical Committee on Railroad Systems and Applications of the IEEE Intelligent Transportation Systems Society. She serves as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *IEEE Intelligent Transportation Systems Magazine*, and *Journal of Intelligent and Robotic Systems*.