Transportation Research Part C 119 (2020) 102680

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Train trajectory optimization for improved on-time arrival under parametric uncertainty

Pengling Wang[a,*], Alessio Trivella[a], Rob M.P. Goverde[b], Francesco Corman[a]

[a] IVT-Institute for Transport Planning and Systems, ETH Zurich, Zurich, Switzerland
[b] Department of Transport and Planning, Delft University Technology, Delft, the Netherlands

ARTICLE INFO

Keywords:
Train trajectory optimization
Parametric uncertainty
Approximate dynamic programming

ABSTRACT

In this paper we study the problem of computing train trajectories in an uncertain environment in which the values of some system parameters are difficult to determine. Specifically, we consider uncertainty in traction force and train resistance, and their impact on travel time and energy consumption. Our ultimate goal is to be able to control trains such that they will arrive on-time, i.e. within the planned running time, regardless of uncertain factors affecting their dynamic or kinematic performance. We formulate the problem as a Markov decision process and solve it using a novel numerical approach which combines: (i) an off-line approximate dynamic programming (ADP) method to learn the energy and time costs over iterations, and (ii) an on-line search process to determine energy-efficient driving strategies that respect the real-time time windows, more in general expressed as train path envelope constraints. To evaluate the performance of our approach, we conducted a numerical study using real-life railway infrastructure and train data. Compared to a set of benchmark driving strategies, the trajectories from our ADP-based method reduce the probability of delayed arrival, and at the same time are able to better use the available running time for energy saving. Our results show that accounting for uncertainty is relevant when computing train trajectories and that our ADP-based method can handle this uncertainty effectively.

1. Introduction

Train operations are subject to several uncertain factors which include bad weather, delays, and mechanical problems, for instance. These factors represent a challenge for drivers in moving the trains in a punctual and energy-efficient manner. One way of helping drivers in this task is by determining energy-efficient train trajectories to guide/drive a train through its scheduled route between two stations within a predefined time with consideration of uncertain factors. Similar issues are found in the design and implementation of driver advisory systems (DAS) automatic train operation (ATO) as automated systems have the possibility to guarantee even higher on-time performance, by adjusting the speed profile and giving advises in real time, as uncertainty manifests.

The problem of finding the specific trajectory, i.e. a sequence of speed values along a time and space axis, arises as an optimal control problem subject to certain operational, geographic, and physical constraints. We refer to this optimization problem as the *train trajectory optimization problem* (TTOP). Typical goals of TTOP considered so far relate to (i) system regularity, by enabling and improving punctual on-time arrival, i.e. running time deviation from a given number must be minimized; (ii) smallest travel time for the person or goods transported, i.e. the travel time must be minimized, and speed must be as large as possible; and (iii) minimum

* Corresponding author.
E-mail addresses: pengling.wang@ivt.baug.ethz.ch (P. Wang), alessio.trivella@ivt.baug.ethz.ch (A. Trivella), R.M.P.Goverde@tudelft.nl (R.M.P. Goverde), francesco.corman@ivt.baug.ethz.ch (F. Corman).

https://doi.org/10.1016/j.trc.2020.102680
Received 7 January 2020; Received in revised form 25 May 2020; Accepted 28 May 2020
Available online 01 September 2020
0968-090X/ © 2020 The Authors. Published by Elsevier Ltd.

energy, i.e. the total energy over the trajectory must be minimized. As these three goals are in conflict, much research is addressing some combination or trade-off of them, or includes one of them as constraint, and another one as objective.

From a mathematical point of view, the minimization of energy given a running time budget is the aspect that received most academic interest. In fact, research on the TTOP started in the 1960s and recent comprehensive reviews on this problem can be found in Yang et al. (2016b),Scheepmaker et al. (2017), and Yin et al. (2017). A classic TTOP focuses on moving the train from one stop to another within a given running time and reducing the energy consumption. The shape of an energy-efficient speed profile has been widely studied by using the Pontryagin's maximum principle, a typical so-called *indirect method* that converts the TTOP into a boundary-value problem using differential equations. It is well-known that for a long journey on a flat track with sufficient running time supplement, the optimal train control strategy consists of the sequence of four control regimes: maximum traction force (MT)–speedholding by partial traction force (SH)–coasting (CO)–maximum braking (MB) (Milroy, 1980; Cheng and Howlett, 1992; Howlett and Pudney, 2012). For a train operating on a track with varying speed limits and gradients, the optimal control strategy was proven to be a sequence of these optimal regimes where the succession of regimes and their switching points also depend on the speed limits and gradients (Pudney and Howlett, 1994; Howlett, 1996; Khmelnitsky, 2000; Liu and Golovitcher, 2003). Finding the optimal switching points is generally a very difficult problem, except for simple cases such as running on a flat track under a single speed limit (Albrecht et al., 2016a; Albrecht et al., 2016b). Different from the indirect method, *direct methods* solve the TTOP by transcribing the optimal control problem to a nonlinear program (NLP), and then solving this program. Wang et al. (2013) proposed a pseudospectral method and a mixed-integer linear programming approach to optimize the speed profile of a single train with the objective of minimizing energy consumption. Wang and Goverde (2016a), Wang and Goverde (2016b), Wang and Goverde (2017) used pseudospectral methods to optimize single-train and multi-train trajectories with consideration of train delays and signal influences. Ye and Liu (2017) directly transcribed the single-train and multi-train trajectory problem to a nonlinear programming (NLP) problem and solved it with existing NLP solvers. Luan et al. (2018a,b) considered the integrated TTOP and real-time traffic management problem using mixed-integer linear and nonlinear programming approaches. The direct methods can usually solve the TTOP without any knowledge of its optimality conditions. However, they often need long computation times to determine the control decision (e.g., more than one minute) and sometimes lead to undesired violent fluctuations in the control profile (Ye and Liu, 2017). Another stream of solution methods for the TTOP is dynamic programming. Ko et al. (2004) formulated the train running process as a multi-stage decision process, and applied a deterministic dynamic programming (DDP) algorithm to search for the optimal control strategy directly. Ghaviha et al. (2017) proposed a DDP approach to find the optimal speed profile for an on-line driver advisory system by taking into account dynamic energy losses in the traction system. Similar to the current paper, they also explicitly distinguish between an off-line learning phase and an on-line phase with smaller memory and computational requirements. Haahr et al. (2017) built a speed profile graph with a set of heuristic rules, and developed a DDP approach to find energy-efficient speed profiles with respect to time constraints, speed restrictions, and passage points. This latter approach is not guaranteed to be optimal, unless all necessary states and partial speed profiles are generated. Zhou et al. (2017) presented a space–time-velocity grid for multiple train operation and tackled the problem using a heuristic DDP. In general, deterministic dynamic programming has been widely used to solve the TTOP as it can find solutions from scratch within practically acceptable computational time even when the method is applied to real-life complicated running conditions.

In addition to the constraints on time of departure and arrival at the source and destination stations, the TTOP may involve intermediate constraints. In fact, time slots on tracks are often allocated to the train allowing passage through the rail infrastructure, and strict timing is imposed on arrivals and departures to enforce customer satisfaction and a high network utilization (Luan et al., 2017). In this sense, on-time passage is the one that ensures the highest network utilization and capacity (Goverde et al., 2013) A generalization of the concept of time window to flexibly include also speed constraints lead to the concept of *train path envelope* (TPE), first proposed by Albrecht et al. (2013), ON-TIME (2014), and further developed by Wang and Goverde (2016a, 2017) to describe the departure and arrival time, and intermediate pass-through constraints as time and speed. In Wang and Goverde (2016a), Haahr et al. (2017), and Ye and Liu (2017), the train needs to pass specific track locations within specific time/speed windows. Additional time/speed constraints at these track locations must be built for the pseudospectral method (Wang and Goverde, 2016a) and the nonlinear programming method (Ye and Liu, 2017) to satisfy the TPE. The dynamic programming approach (Haahr et al., 2017) instead requires defining additional braking/accelerating speed curves around these track locations.

Overall, the scientific literature on the single-train trajectory optimization with intermediate constraints is quite mature. However, not much work has accounted for uncertainty in train control. Yin et al. (2014) addressed the train speed control problem by building expert rules and using a reinforcement learning algorithm to find energy-efficient speed profiles. This approach is capable of using expert experiences to cope with operational uncertainties, for example, accidents and delays. Yang et al. (2016a) developed a stochastic two-stage programming model for integrated timetabling and TTOP that takes into account uncertainty in the train mass as a set of discrete scenarios. Using a genetic algorithm, the authors were able to solve the timetabling problem for a set of train runs in a metro system. Ghaviha et al. (2017) took into account the dynamic power losses and proposed an enhanced model for trajectory optimization including a realistic approximation of the power losses. Apart from these studies, not much attention has been paid to the TTOP considering parametric uncertainty. Most approaches for the TTOP are model-based and rely on a mathematical representation of the considered system. Parameters such as the train resistance and traction/braking force are often estimated based on experience or historical data and may be different from the real-life values (Bešinović et al., 2013; De Martinis and Corman, 2018; Somaschini et al., 2018). Optimizing the train trajectory with inaccurate estimates might lead to suboptimal and even infeasible solutions. Moreover, the existing train trajectory calculations rely on deterministic optimization techniques which cannot handle random variables nor explicitly account for stochastic constraints (Scheepmaker et al., 2017). This leads to a lack of trust in the optimized trajectory, which might discourage the development and application of such trajectory optimization methods in DAS or ATO systems (Panou et al., 2013).

This paper aims to solve the TTOP in the presence of uncertainty, more specifically, with consideration of the stochastic changes of traction force and train resistance, and both including the goal of punctual on-time arrival and minimal energy consumption. We resolve the trade-off between those two performance objectives by defining a strict priority of the punctuality aspect against the energy efficient aspect; that means, we are looking for trajectories that lead to a punctual arrival and, within all those trajectories with the same arrival time, we take the one of minimum energy consumption.

We formulate the train control problem as a Markov decision process (MDP) and tackle this MDP by developing a novel approach that combines approximate dynamic programming (ADP; Powell (2007)) with an on-line search process. ADP and related methods such as reinforcement learning have been used to approximate and solve intractable sequential decision making problems under uncertainty, including high-dimensional MDPs. By approximating the complexity of the optimization problem, ADP methods overcome the curses of dimensionality arising in these problems (Powell, 2007), allowing to deal with large scale programs still within a reasonable and actionable time horizon of computation. This approximation power can also be used to tackle particularly challenging equation structures such as non-linear functions. ADP has the inherent property of separating off-line information such as parameters, approximate relations between variables which can be learned off-line, from some online information regarding the current state. Most notably, ADP is used to tackle in an explicit or an implicit way the complexity that stems from the uncertainty, i.e., which action to take when some variables or parameters are only characterized by a probability distribution and not by a crisp number. ADP has been successfully applied to decrease complexity from locomotive optimization over long time horizon under uncertainty (Powell and Bouzaiene-Ayari, 2007); and optimization of heavy-haul train control subject to uncertain factors such as wind gust and weather condition, which affect the dynamics of a train (Wang et al., 2017), with the goal to minimize forces along a train. In the optimization of timetable rescheduling for the reduction of train delays, ADP has been recently applied to reduce the computational complexity (Ghasempour and Heydecker, 2019); to approximate non-linear constraints and objectives related to energy, dynamics, or passenger choices (Liu et al., 2018); and finally, to incorporate decisions on speed control, train delay, and passenger flow subject to stochastic factors, also including knowledge inference from practitioners (Yin et al., 2016; Yin et al., 2014). ADP-based methods have not yet been applied to TTOP to cope with parametric uncertainty in traction force and train resistance, deliver very fast computation times, and outperform benchmarks based on static or deterministic methods, which is the goal of the present paper.

Different from DDP methods that use deterministic values of the train characteristics and state, such as energy and time, the ADP method we develop learns the performance function and provides a value function approximation, i.e. a label to each possible reachable state. The on-line search phase then uses this value function approximation and updates in real-time the trajectory with consideration of the performance objectives, the past realized trajectory, and the expected values of the future conditions. We numerically evaluate our two-phase ADP-based approach on several instances, based on real-life infrastructure and train data of a Dutch railway corridor. We compare our ADP-based method with a set of benchmarks: (i) a MinTime strategy which aims to run as fast as possible, (ii) a Static speed profile which controls the train with a precomputed sequence of control regimes regardless of the real-time uncertainty and train speed, and (iii) a DDP-based approach which does not consider future uncertainty in parameters explicitly but can dynamically adapt to the realization of this uncertainty. We found that the train trajectories computed with the ADP and benchmarks exhibit different performance, with the former trajectories outperforming the latter trajectories in terms of on-time arrivals, and moreover achieving the minimal energy consumption when the on-time arrival performance is similar. In fact, ADP-based trajectories result in significantly fewer delays and at the same time a better use of the available running time for reducing energy consumption, mitigating the impact of the uncertainty from exogenous factors.

The main contributions of this paper with respect to the TTOP literature can be summarized as follows:

1. Modeling. An MDP is formulated for train trajectory optimization under uncertainty in traction effort and train resistance, which is a new model in the literature. As discussed, the existing research has only considered parametric uncertainty and dynamics in real-time TTOP limitedly.
2. Quality. A two-phase solution methodology is proposed to find energy-efficient driving strategies that respect real-time TPE constraints: it combines an off-line ADP learning process with an on-line search that updates the policies in real-time. This method was shown to outperform the Static speed profile and the DDP-based approach.
3. Practice. Our two-phase solution method can be practically relevant when designing DAS or ATO systems since the computational time needed to compute a real-time driving advice is in the order of a few milliseconds. Most time-consuming calculations are in fact executed off-line and the amount of on-line calculations is limited to looking up a table.

The rest of this paper is organized as follows. In Section 2, we present the classic TTOP model and formulate it as an MDP. In Section 3 and 4 we discuss, respectively, our ADP-based method and the other benchmarks to derive train driving strategies. In Section 5, we present an illustrative example to highlight the differences among these methods. In Section 6, we evaluate the performance of the ADP-based approach and benchmarks on real-life instances. In Section 7, we draw conclusions and discuss future research directions.

2. Model of train control under uncertainty

In this section, we present a model for TTOP under uncertainty. We start in Section 2.1 by introducing the classic train trajectory optimization model. In Section 2.2, we discuss how uncertainty factors affects this model. Finally, we formulate the sequential control problem as an MDP in Section 2.3.

2.1. Classic train trajectory optimization model

The movement of a railway vehicle is determined by a set of physical constraints such as the timetable, speed limits, and other vehicle-related factors. The general train motion equations can be written as follows (Hansen and Pachl, 2014; Wang and Goverde, 2016a):

$$\frac{dv(s)}{ds} = \frac{f(s) - R^{\text{train}}(v) - R^{\text{line}}(s)}{\rho \cdot m \cdot v(s)}, \tag{1}$$

$$\frac{dt(s)}{ds} = \frac{1}{v(s)}, \tag{2}$$

where s is the traversed path [m], $v(s)$ the train velocity [m/s], ρ the rotating mass factor, m the train mass [t], $f(s)$ the force applied on the train [kN], $R^{\text{train}}(v)$ the train resistance force [kN], $R^{\text{line}}(s)$ the line resistance force [kN], and $t(s)$ the traversed time [s]. Distance is chosen as the independent variable because gradients and speed limits occur as functions of distance rather than of time. The train resistance $R^{\text{train}}(v)$ comprises rolling, bearing, dynamic and wind resistances (Pachl, 2002), and can be described as

$$R^{\text{train}}(v) = \alpha + \beta \cdot v + \gamma \cdot v^2,$$

where α, β, and γ are empirically determined coefficients. The line resistance $R^{\text{line}}(s)$ is a function of position and consists of two components: the grade resistance and the curve resistance.

The train traction and braking force are limited by the adhesion between the wheels and the rails as well as the maximum power that can be produced by the engine, that is,

$$-B^{\max} \leqslant f(s) \leqslant \min\left\{F^{\max}, \frac{P^{\max}}{v(s)}\right\}, \tag{3}$$

where $F^{\max}$, $B^{\max}$, and $P^{\max}$ are the upper bounds on traction force, braking force, and traction power, respectively. The train speed cannot exceed the speed limits, i.e.,

$$0 \leqslant v(s) \leqslant V^{\max}(s), \tag{4}$$

where $V^{\max}(s)$ is the train speed limit at position s, including static and temporary speed restrictions.

For a train running between two stops K_0 and K_f, the timetable restricts the departure and arrival time. Moreover, the passing-through times at non-stop stations/junctions should stay within specific time windows to avoid influences on other trains. We use a TPE to describe the departure, arrival, and intermediate pass-through constraints (Albrecht et al., 2013; Wang and Goverde, 2016a). For a mathematical description, we classify a train event at a location (e.g., station or junction) into three types: arrival, departure, and pass-through; we thus define an event set $\mathcal{E} = \{\text{arrival}, \text{departure}, \text{pass} - \text{through}\}$. The TPE for a train is formulated as a set $\mathcal{Z} = \{1, 2, ..., Z\}$ of TPE points, where each point $z \in \mathcal{Z}$ is defined as

$$z = (k_z, e_z, [t_z^{\min}, t_z^{\max}], [v_z^{\min}, v_z^{\max}]), \tag{5}$$

where $k_z \in [K_0, K_f]$ is the location of TPE point z, $e_z \in \mathcal{E}$ is the event, and $t_z^{\min}$, $t_z^{\max}$, $v_z^{\min}$ and $v_z^{\max}$ are respectively the lower and upper bounds on time and speed for TPE point z.

The energy consumption between the stations K_0 and K_f can be written as

$$J = \int_{K_0}^{K_f} f^+(s)ds, \tag{6}$$

where $f^+(s) := \max\{f(s), 0\}$. Traditionally, the most common TTOP consists in finding a sequence of traction and braking force controls that minimizes the energy function (6) subject to constraints (1)–(5). In the current paper instead we follow the complementary approach, that is, to find a sequence of traction and braking force controls that minimizes the deviation from TPE given (1)–(5), and in a second instance, that minimizes the energy function (6).

2.2. Stochastic factors

The classic model of train motion presented in Section 2.1 uses a deterministic representation of the train parameters. However, some of these parameters cannot be easily determined and/or they can change within the travel (Pachl, 2002; De Martinis and Corman, 2018). In particular:

- The train mass, m, includes the mass of cars and locomotives and the mass of loads (passengers/goods). The different amount of passengers/goods at different inter-stations and at different operational periods makes the train mass uncertain (Yang et al., 2016a).
- The braking effort is affected by the friction between the brake blocks and wheel tread, while the friction in turn is influenced by the train speed as well as variable factors including weather conditions (e.g., if the track is wet) and the area of the friction material. In practice, railway companies estimate a maximum braking force $B^{\max}$ for a vehicle from design data or from the results of braking distance tests. This parameter is widely used as a constant in computing train trajectories and estimating braking distances but the variability of the braking force is ignored.
- The traction effort a locomotive can exert is limited by two factors: the maximum force that can be transmitted by adhesion between the wheels and the rail, $F^{\max}$, and the maximum force that can be produced by the traction engine, $P^{\max}/v(s)$. In real-world railway systems, $F^{\max}$ is affected by the adhesion between the wheels and the rail, which is uncertain and cannot be easily

determined (Voltr, 2017). Moreover, $P^{\max}$ is affected by the real-time voltage and current of the rail traction power system, which vary with the number of trains within the same power supply zone (Miyatake and Ko, 2010).
- The train resistance $R^{\text{train}}(v)$ comprises rolling, bearing, dynamic, and wind resistances (Pachl, 2002). It is complicated to find a mathematical model to exactly describe these partial resistances. Thus, the train resistance is not derived from its components but is determined by empirical experiments. Based on the formula $R^{\text{train}}(v) = \alpha + \beta \cdot v + \gamma \cdot v^2$, railway companies have established lists of approximations to calculate the train resistances of their own various types of trains. Still, these approximations might be inaccurate, especially in case of extreme weather conditions as strong wind or snow (see Trivella et al. (2020), for a train resistance equation that models the effect of wind on the train).

Despite all four parameters listed above embedding some degree of uncertainty, this uncertainty is different in the way it evolves and realizes. Moreover, accounting for uncertainty in these four parameters is not equally important in practical applications.

The train mass is related to the number of passengers/amount of goods and is a constant value between two consecutive stops, that is, it does not change until new passengers/goods are loaded/unloaded in the next stop (Yang et al., 2016a). If we could employ sensors on the tracks to weigh the train's carriages, then we could adjust the optimal train trajectory according to the real-time weight. Indeed such weight sensors are being used by railway companies (e.g., NS in the Netherlands) to provide seat information to passengers. In other words, existing technology can be used to remove uncertainty in train mass, which is hence less critical.

Differently from train mass, the other three parameters change dynamically while the train is moving, depending e.g. on the friction between the brake blocks and wheel tread, the power supply system, and weather conditions. The uncertainty in braking force impacts the braking distance. Nevertheless, due to the automatic train protection system, the train should be able to brake/stop in time in all circumstances. In real-life applications, it is always good to leave some safe margins before decreasing speed limits and stops in order to ensure train safety. Therefore, accounting for uncertainty in braking force seems also less critical. Consequently, in this work we focus on uncertainty in traction effort and train resistance, which affect the train running time and the energy consumption.

2.3. Markov decision process of train control

In this section, we formulate the train driving process as an MDP that takes into account the stochastic changes of traction effort and train resistance, and solve it with dynamic programming approaches in order to find the succession of control regimes and their switching points. We discretize the train journey by defining a set of discrete locations (which correspond to the control points) according to three rules:

1. *Critical points of speed limits and gradients.* Both speed limits and gradients are piece-wise constant with respect to the location. We insert a discrete location at each point where either the speed limit or the gradient changes. As a result, speed limits and gradients are constant within each segment.
2. *TPE points.* We insert discrete locations at each TPE point location k_z, for $z \in \mathcal{Z}$, i.e. locations in which the train is subject to a time or speed limitation.
3. *Signal positions.* The signaling system consists of a series of railway signals that divide a railway line into a series of sections, or "blocks", which are important elements in managing train movements. We insert discrete location points corresponding to signal points.

These discrete locations define a set of space segments. Long segments are further divided into multiple smaller segments in order to increase the accuracy of the discretized model, that is, to capture more potential control switching points. Specifically, we define a maximum segment length L^{ref}. If a segment has length $L > L^{\text{ref}}$, then it is further divided into $\lceil L/L^{\text{ref}} \rceil$ segments of equal length, where $\lceil L/L^{\text{ref}} \rceil$ is the smallest integer no smaller than L/L^{ref}. L^{ref} is the length of the longest possible segment in the discretized train journey. Fig. 1 illustrates an example of space discretization.

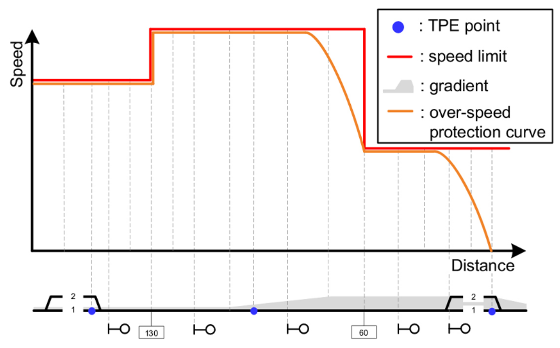


Fig. 1. Example of space discretization.

We now define the TTOP under uncertainty as a finite-horizon MDP. In the following, we characterize the elements that constitute the MDP, i.e., stages, states, decisions, exogenous information, transition function, and cost function (Powell, 2007).

Stage: We associate each discretized location obtained as described above with a stage $d \in \mathcal{D} = \{0, 1, \ldots, D\}$ at which train control decisions can be taken, where $\mathcal{D}$ represents the set of all stages and D the last stage in the horizon (i.e., the arrival station). We denote by s_d the location corresponding to stage d.
State: $S_d = v_d \in \mathcal{S}_d$, where v_d is the train speed at stage d. The speed v_d has to be non-negative and is bounded from above by the speed limit in place at location s_d, i.e., the state space at stage d corresponds to the interval $\mathcal{S}_d \in [0, V^{\max}(s_d)]$. We assume that the entire state or part of it can be measured and is available to the online optimization. We do not include travel time as part of the MDP state and we will explain in Section 3.2 how our method accounts for travel time.
Decision: $x_d \in \mathcal{X}_d(S_d)$ represents a control (traction or braking force to apply) taken at stage d from a state $S_d \in \mathcal{S}_d$. To include energy efficiency in our search, we restrict the decision set to the four optimal control regimes for energy-efficient driving: MT, SH, CO, and MB, that is $\mathcal{X}_d(S_d) \subseteq \{\text{MT}, \text{SH}, \text{CO}, \text{MB}\}$, which is the common choice in the literature (Pudney and Howlett, 1994; Albrecht et al., 2016a; Albrecht et al., 2016b). This would mean that the paths considered are only those that have the best energy performance, for any given running times. In some situations, we restrict the choice further using the following intuitive rules:

- From the departure station, perform the decision of MT to get the train outbound.
- Perform the decision of MB in order to stop at the final destination.
- The train speed must always stay below the speed limit, hence, any decision causing over-speed operation should be avoided. To this end, we adopt an over-speed protection curve as illustrated in Fig. 1. For example, if the train is already running at a speed equal to the speed limit, then maximum traction (MT), is inhibited and we can choose among the remaining actions, including coasting (CO) or speedholding (SH) to keep the train speed constant (we assume that the train is able to maintain a constant speed, i.e., using cruise control). The same applies when the train speed is below the speed limit but it would exceed this limit if choosing MT. In case the speed limit decreases, braking (MB) should be performed as late as possible in order to save time and energy. Braking profiles are therefore inserted at region borders whenever the speed limit decreases.
- Decisions causing unnecessary stops and low travel speeds (e.g. lower than 5 km/h) should not occur.

In summary, not all four controls (MT, SH, CO, and MB) can be selected at any state but only those according to the above heuristic rules. Based on these principles, for each state $S_d \in \mathcal{S}_d$ we define the feasible decision set $\mathcal{X}_d(S_d)$ from which we take a decision $x_d \in \mathcal{X}_d(S_d)$ to move/accelerate/decelerate/stop the train.
Exogenous information: W_d. We consider the three stochastic factors discussed in Section 2.2, i.e.:

$$W_d = (\widetilde{F}_d^{\max}, \widetilde{P}_d^{\max}, \widetilde{R}_d^{\text{train}}) \in \mathcal{W}_d,$$

where $\widetilde{F}_d^{\max}$, $\widetilde{P}_d^{\max}$, and $\widetilde{R}_d^{\text{train}}$ refer to the exogenous and stochastic change in the maximum traction force, maximum traction power, and train resistance force, respectively, occurring between $d-1$ and d. The information W_d is unknown at stage $d-1$ and unfolds before stage d, i.e., we use the convention that variables indexed by d are known at stage d. $\mathcal{W}_d \subset \mathbb{R}^3$ is the support set of these stochastic factors. Bešinović et al. (2013), Powell and Palacín (2015), Sabbaghian (2014) and De Martinis and Corman (2018) all show that the parameters of the traction effort and train resistance vary significantly around their default values and derive stochastic distributions, mostly based on bounded gaussian or uniform distributions, for these parameters for one train type (Bešinović et al., 2013). However, no general conclusions can be obtained about values of parameter distributions for other train engines. Therefore, this study assumes the stochastic factors $\widetilde{F}^{\max}(s)$, $\widetilde{P}^{\max}(s)$, and $\widetilde{R}^{\text{train}}(s)$ follow three independent truncated normal distributions. A truncated normal distribution is the probability distribution corresponding to a normal distribution that is bounded either from above or from below or both (in our case both). Let $\mathcal{N}(\mu, \sigma^2)$ be a normal random variable with mean μ and standard deviation σ. A truncated normal distribution $\mathcal{TN}(\mu, \sigma^2, a, b)$ corresponds to $\mathcal{N}(\mu, \sigma^2)$ conditioned to the interval $[a, b]$. Notice that the probability distribution function (pdf) of $\mathcal{TN}(\mu, \sigma^2, a, b)$ integrates to 1 over $[a, b]$, hence it differs from the pdf of $\mathcal{N}(\mu, \sigma^2)$ over the same interval. The three factors thus follow:

$$\widetilde{F}^{\max}(s) \sim \mathcal{TN}(\mu_f, \sigma_f^2, a_f, b_f) \equiv \mathcal{N}(\mu_f, \sigma_f^2) | [a_f, b_f], \tag{7a}$$

$$\widetilde{P}^{\max}(s) \sim \mathcal{TN}(\mu_p, \sigma_p^2, a_p, b_p) \equiv \mathcal{N}(\mu_p, \sigma_p^2) | [a_p, b_p], \tag{7b}$$

$$\widetilde{R}^{\text{train}}(s) \sim \mathcal{TN}(\mu_r, \sigma_r^2, a_r, b_r) \equiv \mathcal{N}(\mu_r, \sigma_r^2) | [a_r, b_r]. \tag{7c}$$

Transition function: ϕ. Applying a decision $x_d \in \mathcal{X}_d(S_d)$ at state S_d results in a transition to a new state

$$S_{d+1} = \phi(S_d, x_d, W_{d+1}) \in \mathcal{S}_{d+1}, \tag{8}$$

where the transition function ϕ consists of the equations describing the evolution from S_d to S_{d+1} after applying decision x_d. In other words, ϕ describes the change of speed occurring from location s_d to s_{d+1} resulting from the chosen control. The details about ϕ for each action x_d can be found in the appendix. Given a state and decision in stage d, the outcome space at $d+1$ is the set of possible states S_{d+1} and its size is driven by the exogenous/random information W_{d+1} received between d and $d+1$. We assume that starting at stage 0, we observe the exogenous information as the sequence W_1, W_2, ..., meaning that states, decisions, and exogenous information evolve as follows:

$$(S_0, x_0, W_1, S_1, x_1, W_2, \ldots, S_d, x_d, W_{d+1}, \ldots, S_D).$$

Cost function: $C_d(S_d, x_d)$ represents the cost of taking decision x_d at state S_d. The exogenous information W_{d+1} could also play a role in the cost function, i.e., this function could be a random variable at stage *d*. In such case, we can write it as $\widehat{C}_{d+1}(S_d, x_d, W_{d+1})$ and simplify the description by letting $C_d(S_d, x_d) := \mathbb{E}[\widehat{C}_{d+1}(S_d, x_d, W_{d+1})]$ (Powell, 2007).
Typically, the cost function in the TTOP can include two components: the time t_d and energy E_d cost:

$$t_d(S_d, x_d) = \mathbb{E}[\psi(S_d, S_{d+1})], \quad E_d(S_d, x_d) = \mathbb{E}[\chi(S_d, x_d, W_{d+1})], \tag{9}$$

where ψ and χ are functions consisting of the equations that describe the time and energy costs incurred in stage $d + 1$ given the state S_d and decision x_d as well as the realization of the uncertain factors W_{d+1}. The details about ψ and χ can be found in the appendix. We denote by π a *policy*, i.e., a collection of decision functions $\{X_d^\pi,\ d \in \mathcal{D}\}$ such that X_d^π associates each state $S_d \in \mathcal{S}_d$ with a feasible control $X_d^\pi(S_d) = x_d \in \mathcal{X}_d(S_d)$. We call Π the set of feasible policies. The policy that minimizes the energy costs is the one solving the following MDP:

$$\min_{\pi\in\Pi} \mathbb{E}\left[\sum_{d=0}^{D-1} E_d(S_d, x_d) \middle| S_0\right]. \tag{10}$$

A discount factor could also be included in (10). In summary, at each stage (i.e. a location) $d \in \mathcal{D}$, the system (i.e. the train) is in a particular state (i.e. a speed) $S_d \in \mathcal{S}_d$ from which we can take a control decision x_d from the feasible set $\mathcal{X}_d(S_d)$. Taking a decision at a state results in time and energy costs and brings the system to a new future state $S_{d+1} \in \mathcal{S}_{d+1}$ with a given probability $\mathbb{P}(S_{d+1}|S_d, x_d)$. The probability $\mathbb{P}(S_{d+1}|S_d, x_d)$ exists because the exogenous information affects the transition function. The problem is to find the best feasible policy, i.e., the feasible policy that minimizes the total expected cost incurred during the decision horizon by applying this policy.

3. A two-phase approximate dynamic programming method

In this section, we design a two-phase approach to approximate the optimal policy to MDP (10). We start in Section 3.1 by formulating our MDP as a stochastic dynamic program. In Section 3.2, we describe the first phase of our solution approach based on an approximate dynamic programming technique. In Section 3.3, we present the second phase that updates the train trajectory in real-time to meet the TPE constraints.

3.1. Stochastic dynamic programming

It is well-known that the optimal policy to (10) can be found in principle by using the Bellman's equations, that is, by solving the following stochastic dynamic program:

$$V_D(S_D) = 0, \quad \forall\ S_D \in \mathcal{S}_D, \tag{11a}$$

$$V_d(S_d) = \min_{x_d\in\mathcal{X}_d(S_d)} \left\{E_d(S_d, x_d) + \sum_{S'\in\mathcal{S}_{d+1}} \mathbb{P}(S_{d+1} = S'|S_d, x_d)V_{d+1}(S')\right\}, \quad \forall\ d \in \mathcal{D}\backslash\{D\}, \forall\ S_d \in \mathcal{S}_d, \tag{11b}$$

where $V_d(\cdot)$ denotes the *value function* at stage $d \in \mathcal{D}$. Intuitively, $V_d(S_d)$ represents the energy costs incurred from state S_d to the end of the horizon when following an optimal policy (time is not considered for now and will be incorporated in the model in Section 3.3). The value function at the terminal stage is set in (11a). Proceeding backward, the value function is defined in (11b) by minimizing the sum of the immediate cost incurred at stage d and the expected value of the next $d + 1$ state. $\mathbb{P}(S_{d+1} = S'|S_d, x_d)$ represents the probability that taking action x_d from state S_d results in state $s_{d+1} = \phi(S_d, x_d, W_{d+1}) = S'$, and the sum over S' is used in (11b) to compute the expected value function. If we knew the exact value functions $V_d(\cdot)$, for $d \in \mathcal{D}$, then we could compute the optimal decision at stage d and state S_d by solving:

$$X_d(S_d) = \operatorname*{argmin}_{\mathrm{x_d}\in\mathcal{X}_\mathrm{d}(\mathrm{S_d})} \left\{\mathrm{E_d}(\mathrm{S_d}, \mathrm{x_d}) + \sum_{\mathrm{S}'\in\mathcal{S}_{\mathrm{d}+1}} \mathbb{P}(\mathrm{S_{d+1}} = \mathrm{S}' \middle| \mathrm{S_d}, \mathrm{x_d})\mathrm{V_{d+1}}(\mathrm{S}')\right\}. \tag{12}$$

However, (11) is difficult to solve directly due to well-known curses of dimensionality which in our case include a continuous state space S_d and a multi-dimensional exogenous information W_d. Thus, in this section we will develop an approximate dynamic programming method to overcome these curses of dimensionality.

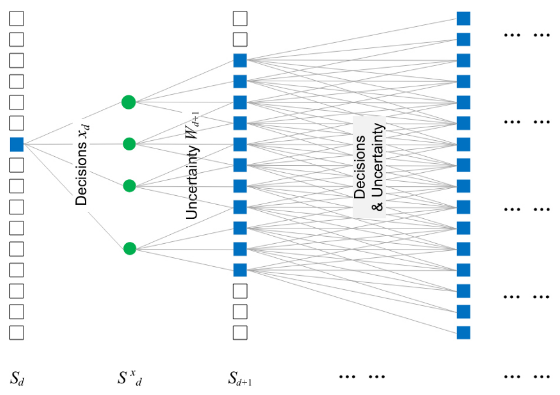


Fig. 2. Example of post-decision states.

To facilitate solving (11), we introduce a *post-decision state* S_d^x, representing the state of the system immediately after decision x_d is made but before the arrival of the new information W_{d+1} (Powell, 2007). This means that states, decisions, and exogenous information evolve as follows (see Fig. 2 for an illustration):

$$(S_0, x_0, S_0^x, W_1, S_1, x_1, S_1^x, W_2, \dots, S_d, x_d, S_d^x, W_{d+1}, \dots, S_D).$$

Using the post-decision state variables, we can reformulate the stochastic dynamic programming equations for $d \in \mathcal{D} \backslash \{D\}$ and $S_d \in \mathcal{S}_d$ as follows (we omit the terminal condition as it is the same as above):

$$V_d^x(S_d^x) = \mathbb{E}[V_{d+1}(S_{d+1}) | S_d^x, W_{d+1}] = \sum_{S' \in \mathcal{S}_{d+1}} \mathbb{P}(S_{d+1} = S' \,|\, S_d, x_d) V_{d+1}(S'), \tag{13a}$$

$$V_d(S_d) = \min_{x_d \in \mathcal{X}_d(S_d)} \{E_d(S_d, x_d) + V_d^x(S_d^x)\}, \tag{13b}$$

where $V_d^x(S_d^x)$ denotes the value function associated with the post-decision state S_d^x. Eq. (13a) assigns an expected downstream cost $\mathbb{E}[V_{d+1}(S_{d+1}) | S_d^x, W_{d+1}]$ to every post-decision state S_d^x, thereby eliminating the need to evaluate all possible outcomes $W_{d+1} \in \mathcal{W}_{d+1}$ for every decision when applying (13b).

In the following, we will develop an ADP algorithm to approximate the post-decision state value functions in (13). If such approximation $\overline{V}_d^x(S_d^x) \simeq V_d^x(S_d^x)$ is available for each d and S_d^x, then we can define a policy π that takes actions as follows:

$$X_d^\pi(S_d) = \operatorname*{argmin}_{x_d \in \mathcal{X}_d(S_d)} \{E_d(S_d, x_d) + \overline{V}_d^x(S_d^x)\}. \tag{14}$$

3.2. Learning phase

To estimate the value functions in (13) as well as approximate the time and energy cost functions, we design a double-pass ADP algorithm. Using a double-pass ADP approach is motivated by recent works that compared this approach with standard single-pass ADP algorithms, showing that policies from the former approach outperform policies from the latter approach (MMes and Riveraes and Rivera, 2017). In other words, double-pass ADP algorithms provide in general better value function approximations.

The ADP learns the value functions iteratively over a number of iterations N. In each iteration n, we generate a *sample path* of the uncertainty, i.e., a particular realization of the exogenous information W_d over the horizon $d = 0, \dots, D$. We call a sample path $w^n \in \Omega$, with $\Omega = \mathcal{W}_1 \times \cdots \times \mathcal{W}_D$ being the set of all possible sample paths. We use the notation $W_d(w^n)$ to indicate the realization at stage d of the sample path w^n in iteration n. At high level, the ADP approach consists in (i) moving forward in the horizon, taking actions based on simulated uncertainty and initial/previous estimates of the value functions, and then (ii) moving backward in the horizon, updating the value functions based on the actions taken in the forward pass.

To highlight that we are dealing with iterations, we add a superscript n to decision and state variables. This means that, at stage d, we are in a state S_d^n and make a decision x_d^n using the value function approximation $\overline{V}_d^{x,n-1}(S_d^n)$. The energy cost and the value functions are indexed with $n-1$ because they are computed using the information from iterations up to $n-1$. The complete procedure is detailed in Algorithm 1.

Algorithm 1. ADP learning phase

```
Inputs: Initial value function approximation $V_d^{x,0}(S_d^x)$, $\forall\ d \in \mathcal{D}$, $S_d^x \in \mathcal{S}_d$; Initial MDP state $S_0^1$; Number of sampling iterations $N$.
For iteration $n = 1$ to $N$ do:
  Step 1. Generate a sample path of uncertainty $w^n$.
  Step 2. Forward pass:
    For $d = 0$ to $D - 1$ do:
      (a) Compute decision $X_d^n(S_d^n) = \underset{x_d^n \in \mathcal{X}_d(S_d^n)}{\operatorname{argmin}} \{E_d^{n-1}(S_d^n, x_d^n) + \overline{V}_d^{x,n-1}(S_d^{x,n})\}$;
      b) Find post-decision state $S_d^{x,n}$ and new pre-decision state $S_{d+1}^n$ with transition functions;
      (c) Compute the observed time and energy cost using $\psi(S_d^n, S_{d+1}^n)$ and $\chi(S_d^n, x_d^n, W_{d+1}(w^n))$.
  Step 3. Backward pass:
    nitialize $\overline{V}_D^{x,n}(S_D^{x,n}) = 0$, $\forall\ S_D^{x,n} \in \mathcal{S}_D$.
    For $d = D - 1$ to 0 do:
      a) Update approximations of time $t_d^n(S_d^n, x_d^n)$ and energy $E_d^n(S_d^n, x_d^n)$ by
         $t_d^n(S_d^n, x_d^n) = \frac{\sum_{n'=0}^{n} \psi(S_d^{n'}, S_{d+1}^{n'})}{n}, \quad E_d^n(S_d^n, x_d^n) = \frac{\sum_{n'=0}^{n} \chi(S_d^{n'}, x_d^{n'}, W_{d+1}(w^{n'}))}{n};$   (15)
      b) Compute $V_d^n(S_d^n) = E_d^n(S_d^n, X_d^{\pi,n}(S_d^n)) + \overline{V}_d^{x,n}(S_d^{x,n})$;
      (c) Compute $\overline{V}_{d-1}^{x,n}(S_{d-1}^{x,n}) = (1-\delta)\overline{V}_{d-1}^{x,n-1}(S_{d-1}^{x,n}) + \delta V_d^n(S_d^n)$.
Outputs: $\forall\ d \in \mathcal{D}$ and sampled state $S_d \in \mathcal{S}_d$: Time cost $t_d^N(S_d^N, x_d^N)$, energy cost $E_d^N(S_d^N, x_d^N)$, value function approximation $\overline{V}_d^{x,N}(S_d^{x,N})$, and action $X_d^N(S_d^N)$.
```

After generating the sample path w^n in Step 1, the algorithm performs the forward pass (Step 2) considering stages from $d = 0$ to $d = D - 1$ sequentially. For a given d, a decision $x_d^n = X_d^{\pi,n}(S_d^n)$ is computed in Step 2(a) based on the value function estimate from the previous iteration $n - 1$. After finding x_d^n, we observe the information $W_{d+1}(w^n)$ to obtain $S_d^{x,n}$ and S_{d+1}^n in Step 2(b). Besides, the time and energy cost incurred when moving from S_d^n to S_d^{n+1}, i.e., $\psi(S_d^n, S_{d+1}^n)$ and $\chi(S_d^n, x_d^n, W_{d+1}(w^n))$, are compute in Step 2(c).

In the backward pass (Step 3), the algorithm sets the terminal value function $\overline{V}_D^{x,n}$ to zero, then proceeds by considering stages backward from $d = D - 1$ to $d = 0$. For each stage d, the approximation for time cost t_d and energy cost E_d is updated in Step 3(a) using, respectively, functions ψ and χ evaluated over the sampling iterations from 1 to n. Indeed Eqs. (15) are sample average approximations of Eqs. (9). Using the updated energy cost and the action from Step 2(a), the value function approximation $V_d^n(\cdot)$ is computed in Step 3(b). Finally, Step 3(c) updates the value function of the post-decision state $\overline{V}_{d-1}^{x,n}(S_{d-1}^{x,n})$ using a convex combination of $\overline{V}_{d-1}^{x,n-1}(S_{d-1}^{x,n})$ and $V_d^n(S_d^n)$, where δ is a weight parameter (MMes and Riveraes and Rivera, 2017).

To summarize, the forward pass sequentially solves a subproblem defined over a sample path of the uncertainty and determines exploitation decisions by moving forward in time. Information on states, decisions, time costs, and energy costs is stored. Afterwards, the time and energy cost estimates are updated in a backward recursion using the information from the forward pass and sample averages, and the value function approximations are computed and updated with a weighting scheme. The process is repeated for a predefined number of iterations where each time a new sample path is generated. Notice that Algorithm 1 requires an initial value function approximation as input in order to make decisions in the first iteration $n = 1$.

3.3. On-line policy update

Algorithm 1 returns a value function approximation and a control policy for energy-efficient train driving without explicit consideration of time, that is, the resulting policy does not necessarily satisfy the TPE constraints nor the scheduled arrival time. We thus propose a second on-line optimization phase in our approach such that the resulting trajectory also fulfills the TPE constraints, which are real-time timing commands from the traffic management system that require quick responses by the DAS/ATO systems. To this end, we use the value functions and cost functions learned while executing Algorithm 1 to update the driving policy in real-time. To ease the notation, in the following we remove superscript N from the outputs of Algorithm 1 and denote them by $t_d(S_d, x_d)$, $E_d(S_d, x_d)$, $\overline{V}_d^x(S_d^x)$, and $X_d^\pi(S_d)$.

Intuitively, our idea is to use the train running time estimates from Algorithm 1 to understand if the energy-efficient driving policy at a given stage complies with the timetable, i.e., if it is expected to fulfill the next TPE constraint or requires instead a variation. In the second case, we accelerate or decelerate the train. Thus, we propose to make a decision at a given state based on the estimated time to reach the next TPE point assuming the train is guided/driven by energy-efficient driving policies. Algorithm 1 provides us with the expected driving time $t_d(S_d, x_d)$ from any state $S_d \in \mathcal{S}_d$ to the successive state S_{d+1} reached when taking action x_d. It also provides us with the energy-efficient driving policy $\pi = \{X_d^\pi,\ d \in \mathcal{D}\}$, which is however not directly usable for now as it does not account for travel time. Using such information, we can estimate the running time needed to drive the train according to π from a state S_d to a TPE point z located later in the horizon, i.e., such that $s_d < k_z$. We denote this time by $T_d^z(S_d)$ and design a backward recursion procedure (Algorithm 2) to estimate it. For each TPE point z, Algorithm 2 considers the stage which is just ahead of the TPE point, then goes through all the previous stages backwardly and computes $T_d^z(S_d)$ with Eq. (16). Algorithm 2 can be seen as a preprocessing phase before we can perform the on-line decision rules that we explain next.

Algorithm 2. Running time estimation

```
Inputs: Time cost functions $t_d$ and energy-efficient policy $\pi$ from Algorithm 1.
For TPE point $z \in \mathcal{Z}$ do:
    (a) Find the stage $D' \in \mathcal{D}$ such that $s_{D'} = k_z$ and let $T^z_{D'}(S_{D'}) = 0,\ \forall\ S_{D'} \in \mathcal{S}_{D'}$;
    (b) For $d = D'$ to 0 do:
        $T^z_d(S_d) = t_d(S_d, X^\pi_d(S_d)) + \sum_{S' \in \mathcal{S}} \mathbb{P}(S_{d+1} = S' \mid S_d, X^\pi_d(S_d)) T^z_{d+1}(S'), \quad \forall\ S_d \in \mathcal{S}_d.$   (16)
Outputs: Driving time estimates $T^z_d(S_d)$ to reach each $k_z$, $z \in \mathcal{Z}$ from each $S_d \in \mathcal{S}_d$, $d \in \mathcal{D}$.
```

We denote by $\pi^c = \{X^{\pi,c}_d,\ d \in \mathcal{D}\}$ an updated driving policy in real-time, where we use the superscript c to distinguish it from the energy-efficient driving policy $\pi = \{X^\pi_d,\ d \in \mathcal{D}\}$ from Algorithm 1. Denote by T^z_{remain} the actual remaining travel time for the train to reach the TPE points $z \in \mathcal{Z}$ located after the current location s_d. The decision at a state S_d should prioritize keeping the estimated running time to reach k_z as close as possible to the actual remaining time T^z_{remain}. So we introduce function (17) to assist decision making and enforce this priority:

$$X^{\pi,c}_d(S_d) = \underset{x_d \in \mathcal{X}_d(S_d)}{\operatorname{argmin}} \left\{ \left| t_d(S_d, x_d) + \sum_{S' \in \mathcal{S}} \mathbb{P}(S_{d+1} = S' \mid S_d, x_d) T^z_{d+1}(S') - T^z_{\text{remain}} \right| \right\}. \tag{17}$$

Generally, learning the value functions and cost functions through iterations in Algorithm 1 might be a time consuming process. However, to minimize the computational time for an on-line application (e.g., a DAS/ATO system), Algorithms 1,2 can be executed off-line. Then, the output from these off-line calculations can be used as input to function (17), which works on-line but is extremely fast as it has to take a single control decision mostly based on pre-computed information. Thus, we can say that our trajectory optimization approach is compatible with real-life applications as DAS/ATO systems from a computational standpoint.

4. Benchmark driving strategies

This section presents a set of benchmark driving policies that are commonly used in the TTOP literature, including a MinTime

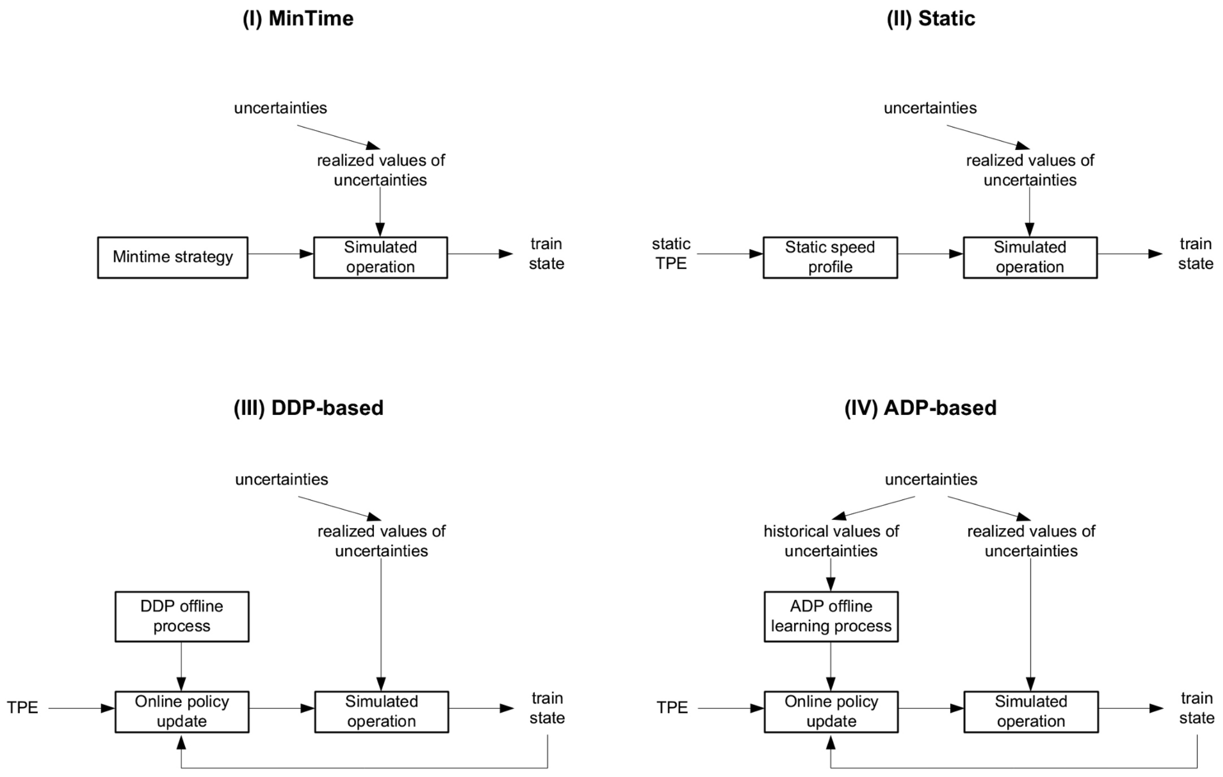


Fig. 3. Architecture of the benchmarks considered.

strategy, a Static speed profile, and a DDP-based approach. We use these policies to evaluate our novel ADP-based approach of Section 3. Fig. 3 describes the four approaches in terms of how the modules and the inputs interact (Corman and Quaglietta, 2015; Corman et al., 2018).

Benchmark I: MinTime strategy (Fig. 3 (I)). It refers to a train trajectory optimization approach that aims at moving the train as fast as possible. Benchmark I does not account for the TPE constraints, often leads to early arrivals and high energy consumption.

Benchmark II: Static speed profile (Fig. 3 (II)). The static speed profile is commonly seen with the application of stand-alone DAS, which has all data downloaded to the train at or prior to the train starting, so that the driver/train is guided/controlled by a static energy-efficient speed profile/policy no matter the changes on real-time traffic states and environment. The static speed profile consists of a static sequence of actions ∈ {MT, SH, CO, MB} leading the train move in an energy-efficient way with respect to a static TPE constraint. The speed profile is deterministic regardless the stochastic changes, real-time TPE and train states.

Benchmark III: DDP-based approach (Fig. 3 (III)). Similar to the ADP-based approach, the DDP-based approach includes two phases as well, the off-line learning phase and the online policy update phase. Moreover, both approaches can observe the current values of the uncertainty and adapt their online decisions according to these values. The difference between DDP-based and ADP-based approach is that in the former method the off-line process optimizes the train trajectory based on deterministic dynamic programming (DDP) that does not account for the uncertainty.

More specifically, removing the uncertainty from the off-line phase means the random variables W_d are treated as zero. The dynamic programming formulation in a deterministic setting is:

$$V_D(S_D) = 0, \quad \forall\ S_D \in \mathcal{S}_D, \tag{18a}$$

$$V_d(S_d) = \min_{x_d \in \mathcal{X}_d(S_d)} \{E_d(S_d, x_d) + V_{d+1}(S_{d+1})\}, \quad \forall\ d \in \mathcal{D}\backslash\{D\}, \forall\ S_d \in \mathcal{S}_d, \tag{18b}$$

where the outcome state S_{d+1} in (18b) is computed with a deterministic transition from the current state S_d and the decision x_d, i.e. $S_{d+1} = \phi(S_d, x_d)$. The recursive step is similar to (11) in structure, but it does not include random variables, that is, cost and transition functions are assumed to be deterministic. If we know the value functions in (18b), then for each stage $d \in \mathcal{D}$ and state $S_d \in \mathcal{S}_D$ the optimal decision is:

$$X_d(S_d) = \underset{\mathrm{x_d} \in \mathcal{X}_\mathrm{d}(\mathrm{S_d})}{\operatorname{argmin}} \{\mathrm{E_d(S_d, x_d) + V_{d+1}(S_{d+1})}\}. \tag{19}$$

To solve (18a), we construct a space-speed network that is formally introduced as a directed acyclic graph $\mathcal{G} = (\mathcal{M}, \mathcal{A})$. The nodes $m \in \mathcal{M}$ in this graph are represented by space-speed pairs $m = (s_d, v_d = S_d)$, defined for each $d \in \mathcal{D}$ (same set of locations used for ADP) and a set of discretized speed points in $\mathcal{S}_d$. The arcs $a \in \mathcal{A}$ are partial speed profiles corresponding to the four driving regimes (MT, SH, CO and MB) that connect two nodes in successive stages d and $d+1$. For consistency, we employ the same rules described in Section 2.3 for the ADP-based method to generate the set of feasible decisions $\mathcal{X}_d(S_d)$ from each node in the graph. This means performing the decision MT from the departure station, MB to stop at the final destination, and MT, SH, CO, or MB in the middle of the journey but avoiding decisions that would violate the speed limits or result in unnecessary stops and low travel speeds. The time and energy costs for each arc $a \in \mathcal{A}$ can be easily computed using the time function ψ and energy function χ (see the appendix for details) by assuming the random variable W_d at stage d equals zero. We use a backward recursion to go through the network and compute the value functions and the energy-efficient driving policy. The process is presented in Algorithm 3.

Algorithm 3. DDP learning phase

```
Input: Space-speed graph $\mathcal{G} = (\mathcal{M}, \mathcal{A})$.
Initialization:(a) Compute time $t_d(S_d, x_d)$ and energy $E_d(S_d, x_d)$ for each arc $a \in \mathcal{A}$;
               (b) Set the terminal value function $V_D(S_D) = 0,\ \forall\ S_D \in \mathcal{S}_D$.
For $d = D - 1$ to 0 do:
    For nodes $m \in \mathcal{M}$ at stage $d$ (i.e., corresponding to location $s_d$) do:
        (a) Solve Eq. (18a) to obtain the value function $V_d(S_d)$;
        (b) Solve Eq. (19) to obtain the decision $X_d(S_d)$;
Outputs: $\forall\ m \in \mathcal{M}$: Value function $V_d(S_d)$ and best decision $X_d(S_d)$.
```

Algorithm 3 computes a deterministic energy-efficient driving policy and can be seen as the equivalent under a deterministic setting of Algorithm 1. Analogously to Algorithm 1, the resulting policy is not guaranteed to satisfy the real-time TPE constraints. Therefore, an online policy update phase is needed by using outputs from Algorithm 3 for the TPE passing time estimation (Algorithm 2) and for the on-line driving policy update. The control decisions depend on the real-time train state, S_d, which is affected by realized values of uncertainties, and the decisions are defined by

$$X_d^{\pi,c}(S_d) = \underset{\mathrm{x_d} \in \mathcal{X}_\mathrm{d}(\mathrm{S_d})}{\operatorname{argmin}} \{|\mathrm{t_d(S_d, x_d) + T^z_{d+1}(S_{d+1}) - T^z_{remain}}|\}. \tag{20}$$

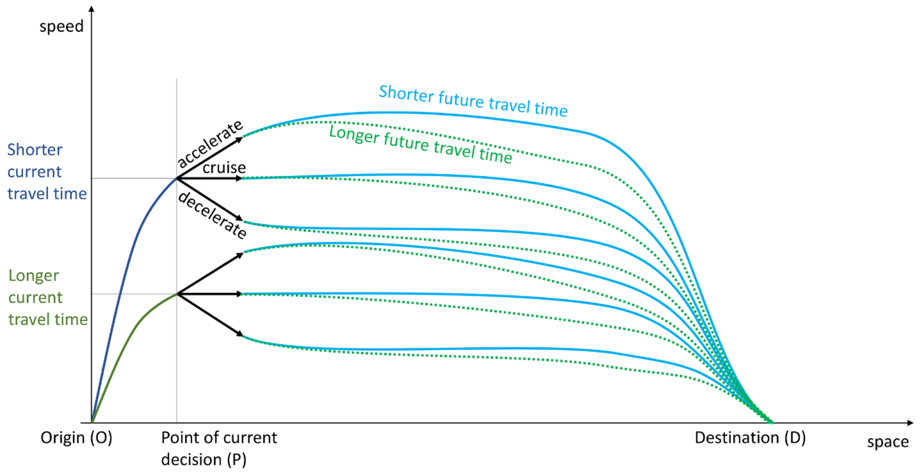

speed
Shorter current travel time
Longer current travel time
accelerate
cruise
decelerate
Shorter future travel time
Longer future travel time
Origin (O)
Point of current decision (P)
Destination (D)
space

Fig. 4. Space-speed diagram for the illustrative example.

5. Illustrative example

To highlight the differences among the methods discussed, we consider an illustrative example displayed in Fig. 4. The figure shows a space (x-axis) speed (y-axis) graph with a train going from an origin O (left of the figure, origin of axes) to a destination D (right of the figure). The train accelerates until a point in space where a decision has to be taken, labeled "point of current decision" (P). Despite the train has accelerated as planned, some stochastic factors resulted in a probability (say 50%) of a higher speed, corresponding to a shorter travel time (say 8 min); and a probability (also consider 50%) of a longer travel time (say 10 min). At the decision point, only three possible choices are considered for simplicity, depicted as three black arrows, namely accelerating ($\neq$ *arrow*), speed holding or cruising (→), or decelerating (↘). After the decision is taken, there is still some distance to cover until destination, over which other stochastic phenomena might occur. Assume for simplicity that there is some probability (say 50%) that the travel time is a bit shorter (say by a minute), and some probability (say 50%) that the travel time is a bit longer (again, for simplicity, by a minute).

The combination of the two possible current travel times (i.e., what happens between the origin O and the decision point P), with the three possible actions (i.e., accelerate, cruise, or decelerate) and the two possible future travel times (from the decision point P to the destination D) gives a grand total of 12 possible cases investigated in Table 1. Assume that the planned travel time is 31 min, and that the goal is to have a travel time smaller than 31 min, but as large as possible due to the wished energy efficiency.

Table 2 reports what the different driving strategies would propose. The first three columns summarize information on uncertainty and decision already introduced in Table 1. For each algorithm, two columns are reported, namely "Eval" is the value for the total travel time that the algorithms evaluates and expects from taking a decision; and "Choice" is the decision taken. Specifically, we use a tick (✓) for the decision identified by the method as feasible and optimal (i.e. chosen); a cross (✗) for a decision that is identified as infeasible; and a hyphen (-) for a decision that is feasible but not optimal (and not chosen).

The MinTime strategy proposes always the same choice, namely accelerating, regardless of the current travel time until the decision point. In fact, no evaluation of the future is needed.

The Static speed profile also proposes always the same choice, regardless of the current travel time. In this case, though, the choice is the one that on average (i.e., over all possible conditions known on beforehand) maximizes the performance function, i.e., arrives the latest possible but on-time. In our specific example, the expected travel time until the decision point is 9, and the expected

Table 1
Decision process and uncertainty of the example. Travel times are reported in minutes.

O → P				P → D			
Expected travel time	Uncertainty scenario	True time	Decision	Expected travel time	Uncertainty scenario	True time	Total time
9	−1 (Pr. 50%)	8	↗	11	−1 (Pr. 50%)	10	18
					+1 (Pr. 50%)	12	20
			→	13	−1 (Pr. 50%)	12	20
					+1 (Pr. 50%)	14	22
			↘	15	−1 (Pr. 50%)	14	22
					+1 (Pr. 50%)	16	24
	+1 (Pr. 50%)	10	↗	19	−1 (Pr. 50%)	18	28
					+1 (Pr. 50%)	20	30
			→	21	−1 (Pr. 50%)	20	30
					+1 (Pr. 50%)	22	32
			↘	23	−1 (Pr. 50%)	22	32
					+1 (Pr. 50%)	24	34

Table 2
Evaluations and choices by the different algorithms.

Time O → P	Decision	Time P → D	MinTime Eval	MinTime Choice	Static Eval	Static Choice	DDP-based Eval	DDP-based Choice	ADP-based Eval	ADP-based Choice
8	↗	10	-	✓	$9+15$	-	$8+11$	-	$8+10$ (Pr. 50%)	-
		12							$8+12$ (Pr. 50%)	
	→	12	-	-	$9+17$	-	$8+13$	-	$8+12$ (Pr. 50%)	-
		14							$8+14$ (Pr. 50%)	
	↘	14	-	-	$9+19$	✓	$8+15$	✓	$8+14$ (Pr. 50%)	✓
		16							$8+16$ (Pr. 50%)	
10	↗	18	-	✓	$9+15$	-	$10+19$	-	$10+18$ (Pr. 50%)	✓
		20							$10+20$ (Pr. 50%)	
	→	20	-	-	$9+17$	-	$10+21$	✓	$10+20$ (Pr. 50%)	✗
		22							$10+22$ (Pr. 50%)	
	↘	22	-	-	$9+19$	✓	$10+23$	✗	$10+22$ (Pr. 50%)	✗
		24							$10+24$ (Pr. 50%)	

remaining travel time after the decision is equal to 15 for acceleration (= [10 + 12 + 18 + 20]/4), 17 for speed holding (= [12 + 14 + 20 + 22]/4), and 19 for deceleration (= [14 + 16 + 22 + 24]/4). This results in an expected travel time for these actions, respectively, of 24 (= 9 + 15), 26 (= 9 + 17), and 28 (= 9 + 19). Since all these times are within the maximum allowed time of 31 min, the slowest is chosen.

The DDP strategy is able to propose different choices based on the current state, that is, based on whether the travel time until the decision point has been shorter (top half of the table) or longer (bottom half of the table) than expected. In each case, the decision is taken by considering the expected travel time until the destination, i.e., the average between the two possible scenarios of shorter/longer travel time towards the destination. These expected values can be determined via dynamic programming as previously discussed. The DDP approach is able to label some decisions as infeasible as they would result in a delay. Within the other choices, the approach chooses the longest travel time possible, which results in decelerating in case of current higher speed or cruising in case of current lower speed.

Finally, ADP also proposes different choices based on the current travel time until the decision point. The decision in ADP is taken considering not only the expected travel time until the destination (as DDP) but the entire probability distribution through the learning process, i.e., ADP is aware of the two possible scenarios of the future uncertainty. As a consequence, ADP can classify as infeasible one additional decision compared to DDP. Specifically, ADP recognises that selecting cruising in case of higher current travel time would result in 50% probability of delay. Therefore, cruising at that particular state is discarded altogether and the best decision is hence to accelerate.

Table 3 reports the aggregate evaluation over all possible cases, taking the decision as detailed in Table 2. The last row of this table reports the average travel time, and the probability of overshooting the target travel time. It is evident how ADP and MinTime are the only two approaches able to be always on-time. However, MinTime does so at the cost of a much shorter travel time, resulting in high energy consumption. The Static approach is unable to adjust the action to the current state, resulting in a 50% probability of delay. DDP is able to adapt the decision to the state, but can only learn expected values and not the precise probability distributions, resulting in 25% infeasibilities. ADP would thus be preferable in this example.

6. Numerical study

In this section, we numerically evaluate the performance of our methods. In Section 6.1, we introduce a set of real railway instances used for our experiments and describe the computational setup. In Section 6.2, we present the numerical results and discuss our findings.

6.1. Instances and computational setup

Our instances are based on the Dutch corridor between Utrecht and ’s-Hertogenbosch. This corridor is a 50 km long double-track line with some multiple-track parts and with traffic in both directions having their own tracks. Eight stations are located along this corridor: Utrecht (Ut), Utrecht Lunetten (Utl), Houten (Htn), Houten Castellum (Htnc), Culemborg (Cl), Geldermalsen (Gdm), Zaltbommel (Zbm) and ’s-Hertogenbosch (Ht). The infrastructural data we have include an accurate description of all track sections, points, speed signs, gradients, and signals over the entire track layout from Utrecht until ’s-Hertogenbosch. We discretize this 50 km long track according to the criteria discussed in Section 2.3 and using $L_{\text{ref}} = 250\text{m}$, that is, the length of the discretized intervals do not exceed this value. We compute trajectories for an intercity running on this corridor in the direction from Ut to Ht. The

Table 3
Performance of the different algorithms.

Case	MinTime	Static	DDP-based	ADP-based
Faster O → P and faster P → D	18	22	22	22
Faster O → P and slower P → D	20	24	24	24
Slower O → P and faster P → D	28	32	30	28
Slower O → P and slower P → D	30	34	32	30
Average total travel time	24	28	27	26
Probability (total travel time > 31)	0%	50%	25%	0%

Table 4
Characteristics of the intercity train.

Parameter	Symbol	Value
Train mass [t]	m	391
Rotating mass factor	ρ	1.06
Train length [m]	–	162
Maximum traction power [kW]	$P^{\max}$	2157
Maximum traction force [kN]	$F^{\max}$	214
Maximum braking rate [m/s^2]	$B^{\max}/m$	0.66
Maximum speed [km/h]	–	140
Train resistance [kN] (v: [km/h])	$R^{\text{train}}(v)$	$5.8584 + 0.0206v + 0.001v^2$

Table 5
TPE time constraints settings. (Unit of time window: min).

Instance	TPE point 1		TPE point 2		TPE point 3	
	location	time	location	time	location	time
(a)	Ut station	[0, 0]	–	–	Ht station	[29, 29]
(b)	Ut station	[0, 0]	Gdm station	[15, 16]	Ht station	[29, 29]

characteristics of the intercity are shown in Table 4 including train mass, rotating mass factor, train length, maximum traction force and power, maximum braking rate, and train resistance. Since the braking rate is the only measurable data characterizing the braking behavior, we set the braking force equal to the product between braking rate and train mass.

To demonstrate the ability of the proposed algorithm to handle time constraints at intermediate locations, we consider two different cases for the TPE constraint, which are displayed in Table 5. Instance (a) assumes that the train runs from Ht to Ut without any intermediate time constraints. The scheduled running time is 29 min. Instances (b) assumes instead that the train operation is subject to one TPE constraint at intermediate station Gdm. Both instances (a) and (b) are feasible, i.e., it is always possible to find speed feasible profiles that respect all TPE constraints as well as the other constraints like speed limits.

The values for the parameters of the distributions (7) could in principle be estimated from real data, such as historical train engines data or historical resistance data under different weather conditions. In this study, such data is not available and we therefore specify three different sets of values, denoted by $W1$, $W2$, and $W3$, that captures different uncertainty behaviours. In $W1$ (red continuous lines in the figure), all three distributions are symmetric with respect to zero, that is, the mean of the truncated normals is zero and the bounding interval is centered in zero too. In $W2$ (black dotted lines in the figure), we instead allow the distributions to have mean different than zero by shifting the distributions left or right compared to $W1$, and keeping the upper and lower bounds symmetric with respect to the peak of the distribution. Specifically, the uncertainty is decreased for the maximum force and power and increased for the train resistance. In $W3$ (blue dashed lines in the figure), we increase the variance of traction force and power, and keep the three distributions symmetric with respect to zero.

Combining the two TPE instances (a)–(b) in Table 5 with the three distribution sets in Fig. 5 gives a total of 6 instances. For each of these instances, we apply the MinTime strategy, Static speed profile, DDP-based and ADP-based methods to derive driving policies. To examine how our driving control adapts to the uncertainty and to evaluate the performance of the algorithms, for each instance we perform 2000 Monte Carlo simulations with the four methods. In each simulation, we (i) generate random changes of traction effort and train resistance through the entire decision horizon based on the distributions in (7), and (ii) run the on-line search process of the four different approaches. The algorithms are implemented in Matlab and executed on a standard laptop equipped with an Intel i7-7600U processor with 16 GB RAM. The ADP and DDP methods involve 180 stages and more than 20000 states. We chose the initial value function approximation used in the ADP-based method equal to the values obtained from the DDP-based method. The off-line phase to approximate the value functions for ADP and DDP took relatively long computational times (roughly 30 min and 2 min,

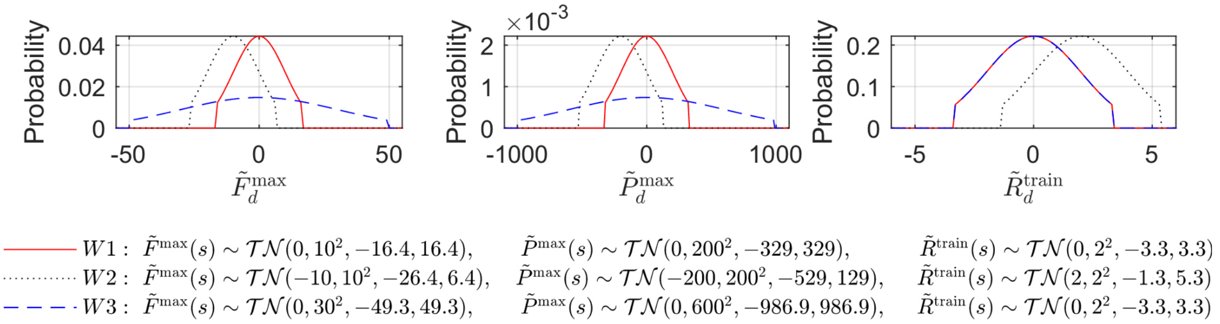


Fig. 5. Probability distribution functions of $\tilde{F}_d^{\max}$ (left), $\tilde{P}_d^{\max}$ (center), and $\tilde{R}_d^{\text{train}}$ (right) in cases $W1$, $W2$ and $W3$.

Table 6
Results. The first and second columns identify the instance. For each method, we report: the average remaining trip time (RTT [s]), the percentage of delays at Ht (Delay [%]), the average delay at Ht (Avg. delay [s]), and the energy consumption (Energy [kWh]).

	MinTime				Static speed profile			
Case	RTT	Delay (%)	Avg. delay	Energy	RTT	Delay (%)	Avg. delay	Energy
$W1$-(a)	106.7	0	0	396.2	-1.1	54.8	6.0	335.8
$W2$-(a)	86.2	0	0	411.7	-82.9	100	82.9	339.2
$W3$-(a)	101.3	0	0	394.7	-4.9	62.0	11.1	334.4
$W1$-(b)	106.7	0	0	396.2	7.7	9.9	3.1	308.1
$W2$-(b)	86.2	0	0	411.7	-49.5	100	49.5	311.2
$W3$-(b)	101.3	0	0	394.7	4.6	28.0	7.0	306.4
Average	98.1	0	0	400.9	-21.0	59.1	26.6	322.5
	DDP-based				**ADP-based**			
Case	RTT	Delay (%)	Avg. delay	Energy	RTT	Delay (%)	Avg. delay	Energy
$W1$-(a)	13.4	0	0	339.6	3.0	0	0	336.4
$W2$-(a)	-28.0	100	28.0	355.5	2.8	4.4	1.6	364.0
$W3$-(a)	29.6	0	0	344.8	2.8	0.2	0	336.5
$W1$-(b)	10.8	0	0	307.2	3.8	0	0	305.2
$W2$-(b)	-17.8	100	17.8	329.2	1.7	2.1	0.5	331.6
$W3$-(b)	15.3	0	0	310.4	3.7	0.1	0.5	304.9
Average	3.9	33.3	7.6	331.1	3.0	1.1	0.4	329.7

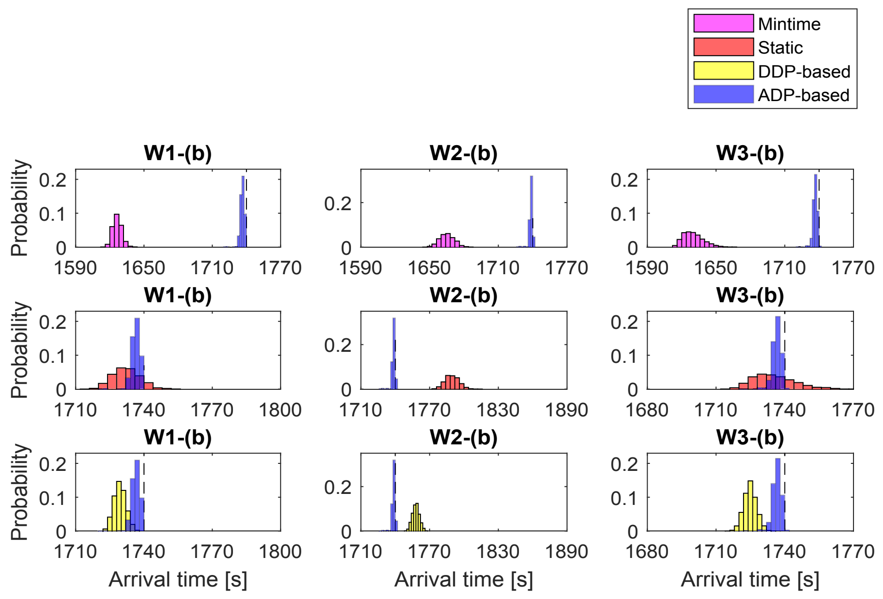
respectively), which can be done off-line with powerful computers. The learning results are subsequently used for on-line policy updating. The on-line phase needs three inputs during the operation: the distance traveled (to determine the stage in the horizon), the current velocity (to determine the state), and the time passed. With this information, the control decision is read from the off-line learning results with function (17). There is no need for data transmission between the on-line phase and the off-line phase during the trip. For the instances we tested, the memory required to store the off-line learning results was only 300–500 kilobytes, and the computational time required by each algorithm to determine an on-line feasible control was about 1–3 ms. Hence, the on-line policy update phase is suitable for real-life train control applications. For more details on developing an on-line DAS for practical use, we refer to Ghaviha et al. (2017), where a complete procedure of designing an on-line DAS on Android devices based on discrete dynamic programming approach is presented and its data requirements discussed. We remark that our on-line phase compares favorably with the computation speed and memory requirements identified by Ghaviha et al. (2017).

6.2. Results

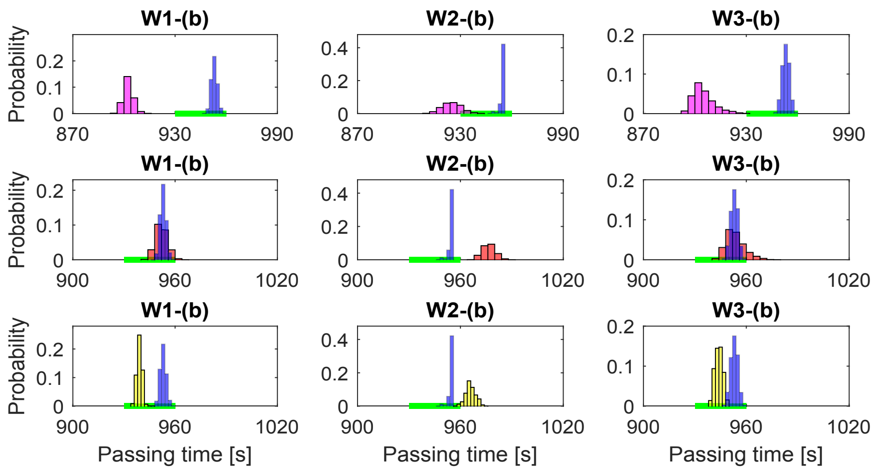
We present our results in this section, where the ADP-based method is compared with the MinTime strategy, Static speed profile and DDP-based methods. In Table 6 we report, for each method and instance, statistics on arrival time, delays, and energy consumption. All numbers in this table represent averages over the 2000 Monte Carlo simulations.

The RTT (remaining trip time) column displayed in this table refers to the difference between the scheduled arrival time at Ht and the simulated arrival time at Ht. Since the developed ADP-based on-line search algorithms treat respecting the timetable as a constraint, most of the resulting simulated train trajectories often arrive at destination earlier than scheduled. With the ADP-based approach, the probability of delay is very small across all instances (0 – 1.6s) and the average RRT is in the range 1.7 – 3.7s. Different from the ADP-based results, with the MinTime strategy, the average RRTs (98.1s on average) and energy consumption (400.9 kWh on average) are relatively high as the MinTime strategy aims to move trains as fast as possible, which leads to early arrival and high energy costs. With the Static speed profile and the DDP-based approach, the probability of delay and RRTs vary substantially depending on the set of stochastic factors employed. Specifically, in case of $W1$ and $W3$, trains guided by the Static speed profile have a high probability of delay (9.9% – 62.0%), while trains guided by the DDP-based policies on average have larger RTTs (10.8 – 29.6s) than the trains guided by the ADP-based policies. In case of $W2$, both the Static speed profile and the DDP-based approaches lead to late arrivals and a very high probability of delay (17.8 – 82.9s late), which are much larger than the corresponding average delays from ADP. To understand this behaviour, recall that the stochastic set $W2$ is characterized by low traction effort (μ_f, $\mu_p < 0$) and a high train resistance ($\mu_r > 0$), and the running time from a state to a TPE point increases on average. The Static and DDP-based driving policies are unable to capture future variability and consequently estimate TPE passing times incorrectly, causing late arrivals. Besides, the Static speed profile leads to bigger delays than the DDP-based approach, because the DDP-based policies decide the optimal actions for each stage based on real-time feedback speed information, while the Static speed profile uses a deterministic precomputed action at each stage.

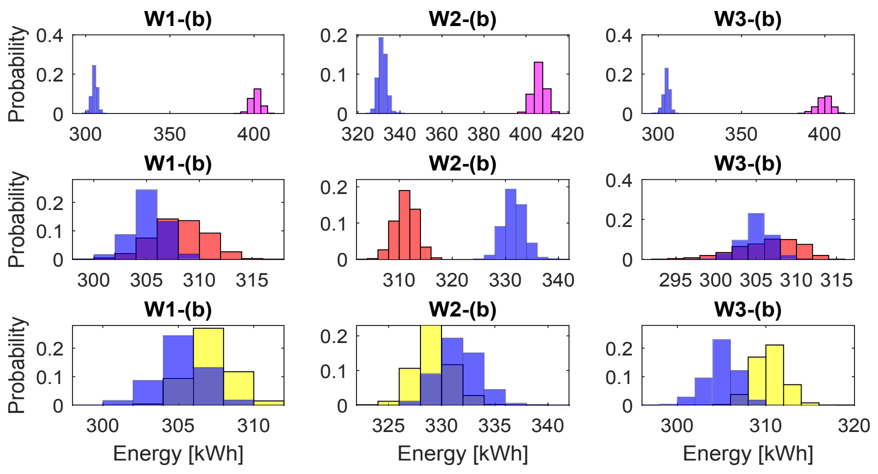
Figs. 6 displays histograms with the probability distributions over 2000 Monte Carlo simulations of arrival times at Ht (top part of the figure), of passing time at Gdm station (middle part of the figure), and of energy costs from Ht to Ht (bottom part of the figure). These figures consider four approaches and instance (b), and complement Table 6 by illustrating full distributions rather than averages alone. As we can see from the top and middle part of Fig. 6, the arrival time distributions from the ADP driving policies are close to the scheduled arrival time (1740s) while only a negligible proportion of these distributions exceed the scheduled time, with very few delays



(a) Arrival time distribution at Ht.



(b) Gdm passing time distribution (green lines refers to the TPE window at Gdm)



(c) Energy distribution

Fig. 6. Probability distributions of arrival times and energy costs.

that never exceed 3 s. The ADP passing time at the intermediate station Gdm respects the TPE time window constraint ([900s, 960s]). This observation implies that our ADP value function approximation and associated real-time driving policies manage travel time in a way that is robust towards different amount of uncertainty. In contrast, the MinTime policies always lead to arrive early at Gdm and Ht, while the arrival time at Gdm and Ht from the Static and DDP driving policies present a substantially larger variability. Especially, the distributions are in fact shifted to the right (delays) for the $W2$ case, compared to the analogous ADP distributions. Thus, when using deterministic methods that neglect uncertainty, it appears harder to handle travel time accurately and respect the schedule, which is critical to operate the railway efficiently. From the bottom part of Fig. 6, we can see that the ADP energy consumption is much less than the energy consumption by applying the MinTime strategy in all the cases (around 11.6% – 23.0% less). The ADP energy consumption distributions are similar with the Static speed profile and DDP-based energy consumption in case of $W1$ and $W3$, with the former slightly outperforming the latter. In case of $W2$, the Static speed profile and DDP energy consumption appear lower than ADP, but at the important cost of generating considerable delays as shown in the arrival time distributions.

7. Conclusion

In this paper, we have studied the problem of optimizing train trajectories under uncertainty in traction force and train resistance, which are parameters that in reality are not known exactly and can vary within the journey. We formulated an MDP for train control that accounts for this uncertainty, to find trajectories guaranteeing the best on time performance (i.e. improving punctuality) despite unknown stochastic variations. We also considered energy efficiency as a second level goal, by restricting the search to only those trajectories which are energy optimal, given any running time budget. We solved the problem by adapting an ADP method based on Monte Carlo simulation and a double-pass framework to learn cost and value function approximations with the objective of minimizing the energy consumption. An on-line search process was designed to use the cost and value functions learned off-line and find on-line driving policies that minimize the deviations from real-time TPE constraints, in other terms that maximize the on-time arrival of trains.

A numerical study was designed using multiple real-life instances with different TPE constraints and uncertainty distributions. The ADP-based method was compared with a set of benchmarks: a MinTime strategy that aims to run the train as fast as possible, a Static speed profile that assumes the train is guided by a static energy-efficient speed profile, and a DDP method that neglects variability in traction force and train resistance but that can react to the different uncertainty realizations when taking a real-time control. Our results showed that control policies resulting from the ADP-based approach outperform those from the three benchmarks. For example, the percentage of delays under ADP is 1.1% on average whereas it presents a large variability (0 – 100%) under the other three methods. The proposed two-phase ADP solution method can be of practical interest in designing DAS or ATO systems not only because of the quality of the resulting trajectories, but also because most calculations are executed off-line and the computational time needed to make an on-line control decision is limited to a few milliseconds. Besides, the proposed algorithm is capable of adjusting the speed profile and giving advises according to both the predefined timetables and the real-time TPE constraints, hence, it could be embedded in stand-alone DASs as well as connected DASs. Future research directions include examining other models for the evolution of the uncertainty, such as stochastic processes, and the calibration of their parameters using data. For example, the weather conditions (wind, rain, snow, etc.) in reality vary in each journey, and evolve dynamically over time. An extension of this work could be modeling this uncertainty as stochastic processes. Besides, the weather conditions have an impact on the train resistance parameters, which in turn can affect the energy consumption and punctuality. Therefore, developing data-driven calibration of those parameters could lead to "weather-aware" train trajectory optimization. Our recent study (Trivella et al., 2020) has built "wind-aware" (but static) train trajectories that exploit the knowledge of wind available before train departure, showing that wind-aware train trajectories present different shape and reduce energy consumption compared to traditional speed profiles computed regardless of any wind information. Finally, this paper focused on the trajectory optimization problem for a single train. Studying the multi-train trajectory optimization problem in an uncertain environment could be a relevant extension.

CRediT authorship contribution statement

Pengling Wang: Conceptualization, Methodology, Software, Writing - original draft. **Alessio Trivella:** Methodology, Validation, Writing - review & editing. **Rob M.P. Goverde:** Supervision, Visualization, Writing - review & editing. **Francesco Corman:** Writing - review & editing.

Acknowledgments

This research was supported by the NWO (Netherlands Organisation for Scientific Research) Rubicon Grant (No: 019.172EN.010); the Swiss National Science Foundation, with Project 181210/DADA; the Swiss Innovation Agency Innosuisse; and the Swiss Competence Center for Energy Research SCCER Mobility.

Appendix A

There are two common ways to compute the transition and cost functions form a particular state S_d to a new state S_{d+1} when taking a decision x_d: (i) by simulating the driving process from S_d to S_{d+1} and calculating these functions during the simulation, and (ii) by using an analytical method. In this appendix we describe this second analytical method.

Consider the train control between two locations s_d to s_{d+1} under a constant speed limit $V_d^{\max}(s) = V_d^{\max}$ and a constant line resistance $R_d^{\text{line}}(s) = R_d^{\text{line}}$. The train can be controlled in an energy-efficient manner by applying one of the four control regimes: MT, SH, CO and MB. The force $f(s)$ exerted to move the train from s_d to s_{d+1} is computed as

$$f(s) = \begin{cases} F_d^{\max} & \text{if } x_d = \text{MT},\\ \alpha + \beta v + \gamma v^2 + R_d^{\text{line}} & \text{if } x_d = \text{SH},\\ 0 & \text{if } x_d = \text{CO},\\ -B_d^{\max} & \text{if } x_d = \text{MB},\end{cases}$$

where $F_d^{\max}$ is the maximum traction force at stage d and is equal to $F_d^{\max} = \min\left\{F^{\max}, \frac{P^{\max}}{v_d}\right\}$, and $B_d^{\max}$ is the maximum braking force at stage d and is equal to $B_d^{\max} = B^{\max}$. From Eq. (1), we can compute

$$\frac{dv}{ds} = \begin{cases} \frac{F_d^{\max} - \alpha - \beta v - \gamma v^2 - R_d^{\text{line}}}{\rho m v} & \text{if } x_d = \text{MT},\\ 0 & \text{if } x_d = \text{SH},\\ \frac{-\alpha - \beta v - \gamma v^2 - R_d^{\text{line}}}{\rho m v} & \text{if } x_d = \text{CO},\\ \frac{-B_d^{\max} - \alpha - \beta v - \gamma v^2 - R_d^{\text{line}}}{\rho m v} & \text{if } x_d = \text{MB}.\end{cases} \tag{21}$$

If we now take exogenous information into account, the expression in (21) becomes

$$\frac{dv}{ds} = \begin{cases} \frac{F_d^{\max} + \widetilde{T}_{d+1}^{\max} - \alpha - \beta v - \gamma v^2 - \widetilde{R}_{d+1}^{\text{train}} - R_d^{\text{line}}}{\rho m v} & \text{if } x_d = \text{MT}\\ 0 & \text{if } x_d = \text{SH}\\ \frac{-\alpha - \beta v - \gamma v^2 - \widetilde{R}_{d+1}^{\text{train}} - R_d^{\text{line}}}{\rho m v} & \text{if } x_d = \text{CO}\\ \frac{-B_d^{\max} - \alpha - \beta v - \gamma v^2 - \widetilde{R}_{d+1}^{\text{train}} - R_d^{\text{line}}}{\rho m v} & \text{if } x_d = \text{MB},\end{cases} \tag{22}$$

where $\widetilde{T}_{d+1}^{\max} = \widetilde{F}_{d+1}^{\max}$ if $F_d^{\max} = F^{\max}$, and $\widetilde{T}_{d+1}^{\max} = \frac{\widetilde{P}_{d+1}^{\max}}{v_d}$ otherwise. We now express Eq. (22) using a more compact formulation as follows:

$$\frac{vdv}{av^2 + bv + c} = -ds, \tag{23}$$

where $a = \begin{cases} \frac{\gamma}{\rho m} & (\text{MT})\\ 0 & (\text{SH})\\ \frac{\gamma}{\rho m} & (\text{CO})\\ \frac{\gamma}{\rho m} & (\text{MB})\end{cases}$, $b = \begin{cases} \frac{\beta}{\rho m} & (\text{MT})\\ 0 & (\text{SH})\\ \frac{\beta}{\rho m} & (\text{CO})\\ \frac{\beta}{\rho m} & (\text{MB})\end{cases}$, and $c = \begin{cases} \frac{-F_d^{\max} - \widetilde{T}_{d+1}^{\max} + \alpha + \widetilde{R}_{d+1}^{\text{train}} + R_d^{\text{line}}}{\rho m} & (\text{MT})\\ 0 & (\text{SH})\\ \frac{\alpha + \widetilde{R}_{d+1}^{\text{train}} + R_d^{\text{line}}}{\rho m} & (\text{CO})\\ \frac{B_d^{\max} + \alpha + \widetilde{R}_{d+1}^{\text{train}} + R_d^{\text{line}}}{\rho m} & (\text{MB})\end{cases}$.

According to common integration rules, we compute:

$$\int \frac{vdv}{av^2 + bv + c} = \frac{1}{2a}\ln\left|av^2 + bv + c\right| - \frac{b}{2a}\int \frac{dv}{av^2 + bv + c},$$

$$\int \frac{dv}{av^2 + bv + c} = \begin{cases} \frac{2}{\sqrt{4ac - b^2}}\arctan\frac{2av + b}{\sqrt{4ac - b^2}} + K & (b^2 < 4ac)\\ \frac{1}{\sqrt{b^2 - 4ac}}\ln\left|\frac{2av + b - \sqrt{b^2 - 4ac}}{2av + b + \sqrt{b^2 - 4ac}}\right| + K & (b^2 > 4ac)\end{cases},$$

where K is a constant. The left-hand side of Eq. (23) can therefore be integrated as

$$\int_{v_d}^{v_{d+1}} \frac{vdv}{av^2 + bv + c} = \begin{cases} \frac{1}{2a}ln\left|av_{d+1}^2 + bv_{d+1} + c\right| - \frac{b}{a\sqrt{4ac - b^2}}arctan\frac{2av_{d+1} + b}{\sqrt{4ac - b^2}} & (b^2 < 4ac)\\ \quad - \left(\frac{1}{2a}ln\left|av_d^2 + bv_d + c\right| - \frac{b}{a\sqrt{4ac - b^2}}arctan\frac{2av_d + b}{\sqrt{4ac - b^2}}\right) & \\ \frac{1}{2a}ln\left|av_{d+1}^2 + bv_{d+1} + c\right| - \frac{b}{2a\sqrt{b^2 - 4ac}}ln\left|\frac{2av_{d+1} + b - \sqrt{b^2 - 4ac}}{2av_{d+1} + b + \sqrt{b^2 - 4ac}}\right| & (b^2 > 4ac)\\ \quad - \left(\frac{1}{2a}ln\left|av_d^2 + bv_d + c\right| - \frac{b}{2a\sqrt{b^2 - 4ac}}ln\left|\frac{2av_d + b - \sqrt{b^2 - 4ac}}{2av_d + b + \sqrt{b^2 - 4ac}}\right|\right) & \end{cases}. \tag{24}$$

while the right-hand side of (23) is simply integrated as

$$\int_{s_d}^{s_{d+1}} - ds = s_d - s_{d+1}. \tag{25}$$

By equating the expressions in (24) and (25), it is possible to express v_{d+1} as a function $v_{d+1} = \phi(S_d, x_d, W_{d+1})$ that describes the evolution of the system from $S_d(=v_d)$ to $S_{d+1}(=v_{d+1})$.

Regarding the travel time from location s_d to s_{d+1}, we can approximate it by using a trapezoidal integration rule (Wang et al., 2013):

$$t = \frac{1}{2}\left(\frac{1}{v_d} + \frac{1}{v_{d+1}}\right)(s_{d+1} - s_d),$$

which can be expressed using a function $t = \psi(S_d, S_{d+1})$.

Finally, the energy consumption from s_d to s_{d+1} is computed with

$$E = \int_{s_d}^{s_{d+1}} max\{f(s), 0\}ds = \begin{cases} (F_d^{max} + \widetilde{F}_{d+1}^{max})\cdot(s_{d+1} - s_d) & if\ x_d = MT \\ (\alpha + \beta v_d + \gamma v_d^2 + \widetilde{R}_{d+1}^{train} + R_d^{line})\cdot(s_{d+1} - s_d) & if\ x_d = SH \\ 0 & if\ x_d = CO \\ 0 & if\ x_d = MB, \end{cases}$$

that we express more compactly using a function $E = \chi(S_d, x_d, W_{d+1})$.

References

Albrecht, A.R., Howlett, P.G., Pudney, P.J., Vu, X., Zhou, P., 2016a. The key principles of optimal train control part 1: formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points. Transp. Res. Part B: Methodol. 94, 482–508.
Albrecht, A.R., Howlett, P.G., Pudney, P.J., Vu, X., Zhou, P., 2016b. The key principles of optimal train control part 2: existence of an optimal strategy, the local energy minimization principle, uniqueness, computational techniques. Transp. Res. Part B: Methodol. 94, 509–538.
Albrecht, T., Binder, A., Gassel, C., 2013. Applications of real-time speed control in rail-bound public transportation systems. IET Intel. Transport Syst. 7 (3), 305–314.
Bešinović, N., Quaglietta, E., Goverde, R.M., 2013. A simulation-based optimization approach for the calibration of dynamic train speed profiles. J. Rail Transp. Plann. Manage. 3 (4), 126–136.
Cheng, J., Howlett, P., 1992. Application of critical velocities to the minimisation of fuel consumption in the control of trains. Automatica 28 (1), 165–169.
Corman, F., Quaglietta, E., 2015. Closing the loop in real-time railway control: Framework design and impacts on operations. Transp. Res. Part C: Emerg. Technol. 54, 15–39.
Corman, F., Quaglietta, E., Goverde, R.M., 2018. Automated real-time railway traffic control: an experimental analysis of reliability, resilience and robustness. Transp. Plann. Technol 41 (4), 421–447.
De Martinis, V., Corman, F., 2018. Data-driven perspectives for energy efficient operations in railway systems: current practices and future opportunities. Transp. Res, Part C: Emerg. Technol. 95, 679–697.
Ghasempour, T., Heydecker, B., 2019. Adaptive railway traffic control using approximate dynamic programming. Transp. Res. Part C: Emerg. Technol.
Ghaviha, N., Bohlin, M., Holmberg, C., Dahlquist, E., Skoglund, R., Jonasson, D., 2017. A driver advisory system with dynamic losses for passenger electric multiple units. Transp. Res. Part C: Emerg. Technol. 85, 111–130.
Goverde, R.M.P., Corman, F., D'Ariano, A., 2013. Railway line capacity consumption of different railway signalling systems under scheduled and disturbed conditions. J. Rail Transp. Plann. Manage. 3 (3), 78–94.
Haahr, J.T., Pisinger, D., Sabbaghian, M., 2017. A dynamic programming approach for optimizing train speed profiles with speed restrictions and passage points. Transp. Res. Part B: Methodol. 99, 167–182.
Hansen, I.A., Pachl, J., 2014. Railway Timetabling & Operations. Eurailpress, Hamburg.
Howlett, P., 1996. Optimal strategies for the control of a train. Automatica 32 (4), 519–532.
Howlett, P., Pudney, P., 2012. Energy-efficient Train Control. Springer.
Khmelnitsky, E., 2000. On an optimal control problem of train operation. IEEE Trans. Autom. Control 45 (7), 1257–1266.
Ko, H., Koseki, T., Miyatake, M., 2004. Application of dynamic programming to the optimization of the running profile of a train. WIT Trans. Built Environ. 74.
Liu, R., Golovitcher, I.M., 2003. Energy-efficient operation of rail vehicles. Transp. Res. Part A: Policy Pract. 37 (10), 917–932.
Liu, R., Li, S., Yang, L., Yin, J., 2018. Energy-efficient subway train scheduling design with time-dependent demand based on an approximate dynamic programming approach. IEEE Trans. Syst. Man Cybernet.: Syst. Article Adv.
Luan, X., Miao, J., Meng, L., Corman, F., Lodewijks, G., 2017. Integrated optimization on train scheduling and preventive maintenance time slots planning. Transp. Res. Part C: Emerg. Technol. 80, 329–359.
Luan, X., Wang, Y., De Schutter, B., Meng, L., Lodewijks, G., Corman, F., 2018a. Integration of real-time traffic management and train control for rail networks-part 1: Optimization problems and solution approaches. Transp. Res. Part B: Methodol. 115, 41–71.
Luan, X., Wang, Y., De Schutter, B., Meng, L., Lodewijks, G., Corman, F., 2018b. Integration of real-time traffic management and train control for rail networks-part 2: Extensions towards energy-efficient train operations. Transp. Res. Part B: Methodol. 115, 72–94.
Mes, M.R., Rivera, A.P., 2017. Approximate dynamic programming by practical examples. In: Markov Decision Processes in Practice. Springer, pp. 63–101.
Milroy, I.P., 1980. Aspects of automatic train control. Ph.D. thesis. Loughborough University, Leicestershire, UK.
Miyatake, M., Ko, H., 2010. Optimization of train speed profile for minimum energy consumption. IEEJ Trans. Electrical Electron. Eng. 5 (3), 263–269.
ON-TIME, 2014. Best practice, recommendations and standardisation. Deliverable ONT-WP01-DEL-003.
Pachl, J., 2002. Railway operation and control. VTD Rail Publishing.
Panou, K., Tzieropoulos, P., Emery, D., 2013. Railway driver advice systems: Evaluation of methods, tools and systems. J. Rail Transp. Plann. Manage. 3 (4), 150–162.
Powell, J., Palacín, R., 2015. A comparison of modelled and real-life driving profiles for the simulation of railway vehicle operation. Transp. Plann. Technol 38 (1), 78–93.
Powell, W., Bouzaiene-Ayari, B., 2007. Approximate dynamic programming for rail operations. In: 7th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'07). pp. 191–208.
Powell, W.B., 2007. Approximate Dynamic Programming: Solving the Curses of Dimensionality, vol. 703 John Wiley & Sons.
Pudney, P., Howlett, P., 1994. Optimal driving strategies for a train journey with speed limits. J. Aust. Math. Soc. Ser. B Appl. Math. 36 (01), 38–49.
Sabbaghian, M., 2014. A stable speed advice for reliable and safe rail traffic. Master's thesis. Univeristy of Twente, the Netherlands.

Scheepmaker, G.M., Goverde, R.M., Kroon, L.G., 2017. Review of energy-efficient train control and timetabling. Eur. J. Oper. Res. 257 (2), 355–376.
Somaschini, C., Argentini, T., Rocchi, D., Schito, P., Tomasini, G., 2018. A new methodology for the assessment of the running resistance of trains without knowing the characteristics of the track: application to full-scale experimental data. Proc. Inst. Mech. Eng., Part F: J. Rail Rapid Transit, 1814–1827.
Trivella, A., Wang, P., Corman, F., 2020. https://doi.org/10.1016/j.ejtl.2020.100013.
Voltr, P., 2017. Calculation of locomotive traction force in transient rolling contact.
Wang, P., Goverde, R.M.P., 2016a. Multiple-phase train trajectory optimization with signalling and operational constraints. Transp. Res. Part C: Emerg. Technol. 69, 255–275.
Wang, P., Goverde, R.M.P., 2016b. Two-train trajectory optimization with a green-wave policy. Transp. Res. Rec. 2546 (1), 112–120.
Wang, P., Goverde, R.M.P., 2017. Multi-train trajectory optimization for energy efficiency and delay recovery on single-track railway lines. Transp. Res. Part B: Methodol. 105, 340–361.
Wang, X., Tang, T., He, H., 2017. Optimal control of heavy haul train based on approximate dynamic programming. Adv. Mech. Eng. 9 (4) 1687814017698110.
Wang, Y., De Schutter, B., van den Boom, T.J., Ning, B., 2013. Optimal trajectory planning for trains–a pseudospectral method and a mixed integer linear programming approach. Transp. Res. Part C: Emerg. Technol. 29, 97–114.
Yang, X., Chen, A., Ning, B., Tang, T., 2016a. A stochastic model for the integrated optimization on metro timetable and speed profile with uncertain train mass. Transp. Res. Part B: Methodol. 91, 424–445.
Yang, X., Li, X., Ning, B., Tang, T., 2016b. A survey on energy-efficient train operation for urban rail transit. IEEE Trans. Intell. Transp. Syst. 17 (1), 2–13.
Ye, H., Liu, R., 2017. Nonlinear programming methods based on closed-form expressions for optimal train control. Transp. Res. Part C: Emerg. Technol. 82, 102–123.
Yin, J., Chen, D., Li, L., 2014. Intelligent train operation algorithms for subway by expert system and reinforcement learning. IEEE Trans. Intell. Transp. Syst. 15 (6), 2561–2571.
Yin, J., Tang, T., Yang, L., Gao, Z., Ran, B., 2016. Energy-efficient metro train rescheduling with uncertain time-variant passenger demands: an approximate dynamic programming approach. Transp. Res. Part B: Methodol. 91, 178–210.
Yin, J., Tang, T., Yang, L., Xun, J., Huang, Y., Gao, Z., 2017. Research and development of automatic train operation for railway transportation systems: a survey. Transp. Res. Part C: Emerg. Technol. 85, 548–572.
Zhou, L., Tong, L.C., Chen, J., Tang, J., Zhou, X., 2017. Joint optimization of high-speed train timetables and speed profiles: a unified modeling approach using space-time-speed grid networks. Transp. Res. Part B: Methodol. 97, 157–181.