


TTOBench—An open-source library for train trajectory optimization

Journal Article**Author(s):**

Kouzoupis, Dimitris; Pendharkar, Ishan; [Corman, Francesco](#) 

Publication date:

2023-09

Permanent link:

<https://doi.org/10.3929/ethz-b-000632317>

Rights / license:

[Creative Commons Attribution 4.0 International](#)

Originally published in:

Operations Research Forum 4(4), <https://doi.org/10.1007/s43069-023-00248-x>

Funding acknowledgement:

180545 - NCCR Automation (phase I) (SNF)



TTOBench—an Open-Source Library for Train Trajectory Optimization

Dimitris Kouzoupis^{1,2} · Ishan Pendharkar¹ · Francesco Corman²

Received: 22 December 2022 / Accepted: 23 August 2023
© The Author(s) 2023

Abstract

Over the past few years, the field of train trajectory optimization has been enriched with numerous publications proposing novel solution algorithms, solving known practical problems or applying existing methods to academic and industrial applications. However, the data that were used in the numerical experiments - especially those related to the railway infrastructure - are rarely public, making the comparison of different algorithms or the reproduction of any results difficult if not impossible. In this work, we try to bridge this gap by introducing a library of track data with slope and speed limit information as a first step towards an open-source benchmark suite for train trajectory optimization. The collected items were either publicly available in some form or acquired by the digitalization of published figures. We encourage authors in the field to contribute to this library for the common benefit of the community.

Keywords Energy-efficient train control · Benchmark library · Track data

1 Introduction

Train trajectory optimization (TTO), often referred to as energy-efficient train control (EETC) or minimum-time train control (MTTC) depending on the objective function of the underlying optimization problem, has been an active topic in the field of transportation research due to its potential to improve driver advisory systems and automatic train operation algorithms. For a recent, comprehensive

This article is part of the Topical Collection on *Public Transport Optimization: From Theory to Practice*

✉ Dimitris Kouzoupis
dimitris.kouzoupis@fhnw.ch

¹ Institute of Electric Power Systems, Fachhochschule Nordwestschweiz (FHNW), Klosterzelgstrasse 2, Windisch 5210, Switzerland

² Department of Civil, Environmental and Geomatic Engineering, ETH Zürich, Stefano-Franscini-Platz 5, Zürich 8093, Switzerland

review the reader is referred to [1]. From early on, numerical methods based on Pontryagin's maximum principle were used to solve the train control problem, translating the calculation of the optimal trajectories into the computation of switching points between optimal driving regimes [2, 3]. One disadvantage of such methods is that small variations in the problem formulation require expert knowledge to update the solution algorithm, justifying the numerous publications on the topic. Moreover, the set of optimal driving regimes is based on the assumption that the efficiency of the propulsion system is constant. Modelling the nonlinear nature of the power losses gives rise to additional driving regimes (e.g., limited acceleration) and the developed algorithms are no longer applicable [4, 5]. On the contrary, solution methods based on the principle of dynamic programming can elegantly treat such nonlinear models [4, 6]. They are well suited for TTO because the typical problem size (number of states and controls) is small enough to avoid the so-called curse of dimensionality that such methods suffer from. Dynamic programming methods require a computationally intensive initialization step, where the state feedback law is calculated for all points on the discretized state space. It is important to note that updates on problem parameters that would require a new initialization may severely affect the applicability of the algorithm. Recently, direct optimal control methods, which transform the train control problem into a nonlinear programming problem (NLP), also attracted the attention of researchers in the field due to the flexibility they allow in the problem formulation and the availability of reliable NLP solvers. Algorithms based on a special variant of direct transcription methods - the pseudospectral method - were proposed in [7–9], while the authors in [10] discretize the optimal control problem using direct multiple shooting.

With so many different methods and software implementations, a convenient framework to compare the performance of different approaches under the same conditions becomes scientifically and practically relevant. In the context of TTO, same conditions imply common track data, train specifications and timetable-related constraints. As the most relevant train and problem data can be typically summarized in a short table, the focus of this paper is on a library of track data, which are either too lengthy to be part of a scientific publication or kept confidential by the railway companies. The library is hosted on GitHub and contributions that follow the indicated guidelines are always welcome [11]. Finally, we note that the recently published library `EETTLIB` [12], which contains a series of benchmark problems for energy efficient train timetabling, is based on fixed power profiles of trains on track segments and does not provide any information on track characteristics such as gradients and speed limits.

The structure of the paper is as follows: Sect. 2 introduces the TTO problem in continuous space. Section 3 explains the current content and structure of the benchmark library. Section 4 compares three TTO algorithms in terms of energy efficiency on the tracks of the database. Finally, Sect. 5 discusses possible extensions of the library and concludes the paper.

2 Train Trajectory Optimization

Although the track database can be of use in several domains of railway research (e.g., in train simulation software), its creation was motivated by the need for common benchmark examples in the field of TTO. In this section we briefly go through one of the many possible problem formulations with a focus on the parameters that are related to the infrastructure.

Two important classifications of problem formulations arise with the choice of the objective function and the independent variable: time versus energy optimization and time versus space discretization. For the sake of clarity, we will briefly present the train control problem in continuous space. Let us choose as differential states of the train dynamics the elapsed time t in s and the square of the velocity $b := v^2$ in m^2/s^2 . As control input we define the specific electrical force f^{el} , where positive values denote traction and negative values regenerative braking. The term specific is used throughout this section to express the division of the corresponding variable by the total mass of the train $m\rho$, with m the train mass in kg and ρ the dimensionless rotating mass factor. This implies that f^{el} is expressed in m/s^2 or equivalently N/kg. The other two forces acting on the train are the specific line resistance force f^{lr} , comprising the sum of forces due track slope and curvature, and the specific rolling resistance force f^{rr} , which is typically expressed as a second degree polynomial of the train speed. Since forces due to curvature are almost always ignored in the literature and there is hardly any data available, they are currently not considered in the benchmark library. Positive values of the line resistance imply uphill track sections and negative values downhill track sections. The rolling resistance force is always positive.

Based on these definitions, the acceleration of the train in m/s is given by Newton's second law of motion: $a(s) := f^{el}(s) - f^{rr}(v(s)) - f^{lr}(s)$, and the TTO problem in continuous space can be formulated as follows:

$$\underset{t(\cdot), b(\cdot), f^{el}(\cdot)}{\text{minimize}} \quad u \int_S^0 f^{el}(s) + \frac{p^{loss}(s)}{v(s)} ds + (1 - u)t(S) \tag{1a}$$

$$\text{subject to } b'(s) = 2 a(s), \quad s \in [0, S], \tag{1b}$$

$$t'(s) = \frac{1}{\sqrt{b(s)}} \quad s \in [0, S], \tag{1c}$$

$$\underline{b}(s) \leq b(s) \leq \bar{b}(s), \quad s \in [0, S], \tag{1d}$$

$$\underline{f}^{el} \leq f^{el}(s) \leq \bar{f}^{el}, \quad s \in [0, S], \tag{1e}$$

$$\underline{p} \leq f^{el}(s)\sqrt{b(s)} \leq \bar{p}, \quad s \in [0, S], \tag{1f}$$

with S the length of the track in m. Equation (1a) describes the objective to be minimized, with p^{loss} the specific power losses in the traction chain and u a binary variable that is equal to 1 for EETC and 0 for MTTC. Equations (1b) and (1c) enforce the train dynamics, Equation (1d) ensures that the speed limits are respected and Equations (1e) and (1f) summarize the physical constraints of the propulsion system with respect to specific electrical force and power.

The track-related information embedded in Equation (1a) consists of: the track length S , the values of the specific line resistance f^{lr} and the speed limits $\bar{v}(s) := \sqrt{\bar{b}(s)}$ along the track. With $\theta(s)$ denoting the angle of the track in rad, the commonly used track gradient $g(s)$ in ‰ is defined as $g(s) := 10^3 \tan(\theta(s))$ and the value of the specific line resistance is given by:

$$f^{\text{lr}}(s) = \frac{g}{\rho} \frac{g(s)}{10^3}, \quad (2)$$

with g the gravitational acceleration in m/s^2 . Equation (2) assumes that for small slopes it holds $\tan(\theta(s)) \approx \sin(\theta(s))$ and (as already mentioned) neglects forces due to curvature. Typically, track data consist of track sections with piece-wise constant gradient, similarly to speed limit values.

Remark 1 Most TTO algorithms assume a point mass model of the train. To avoid speed limit violations, the length of the vehicle should be used in a pre-processing step to transform the speed limits over the track accordingly. Similarly, the track gradients should be adjusted based on the mass distribution of the train [3].

3 Open-source Benchmark Library

In this section, we explain in detail how each track in TTOBench is defined, how the database was created and how it is structured.

3.1 Track Description

Every track in the library is currently defined by a JSON file with the following fields: *metadata*, *altitude*, *stops*, *speed limits* and *gradients*. Future extensions of the benchmark library may include further properties, such as curvature information, track gauge or tunnel resistance, which are for the moment not considered due to lack of public data.

- *metadata*: a dictionary of key-value pairs with general information on the file content. Required fields for a valid track file are *id* and *library version*. The *id* is a unique identifier that should contain only letters, numbers and underscores. If applicable, the first part of the *id* followed by an underscore shall indicate the country of origin via the ISO Country Code. Otherwise, the prefix 00 shall be used. The *library version* is necessary for compatibility purposes, in case there are changes in the library format. The optional fields at the time of writing are: *description* (information about the

track in plain text), *created by* (author of the file) and *license* (to indicate the terms of use in case the file is distributed outside the library).

- *altitude*: altitude at beginning of track. This field is optional and may be used in combination with the gradients to plot the altitude profile of the complete track.
- *stops*: a list of positions on track where stops are located. The first element of the list must be equal to 0 and the remaining positions should follow a strictly increasing order. The last element of the list indicates the length of the track S . Optimized trajectories may be derived from any departure to any destination point in the list.
- *speed limits*: a list of pairs - position, speed limit - indicating the beginnings of track sections and their respective velocity restrictions. The positions must be strictly increasing and every speed limit should be different than its predecessor (otherwise there would be no need to add this pair in the list). The position of the first pair should be 0 and the position of the last pair cannot exceed the length of the track.
- *gradients*: a list of pairs - position, gradient - indicating the beginnings of track sections and their respective (approximately constant) slope. The positions must be strictly increasing and every gradient should be different than its predecessor. Positive and negative gradient values indicate uphill and downhill sections respectively. The position of the first pair should be 0 and the position of the last pair cannot exceed the length of the track. The field can be omitted for level tracks.

To improve the readability and ease the interpretation of the track files, every field (except for the *metadata*) contains all the information about the units. Currently, the positions are expressed in m, the speed limits in km/h and the gradients in ‰. Note that the positions of speed limits and gradients will in general not coincide and a preprocessing step to derive track sections with constant properties is necessary in the context of trajectory optimization. This representation, however, makes a clear distinction between points with a speed limit change and points with a slope change and simplifies the interpretation of the track properties. The details of the JSON files are summarized in Table 1 and a simple example of a JSON file is provided in Appendix. The use of km/h over the SI unit of m/s is adopted to ensure that the speed limits can be represented by integer values.

Table 1 Properties of tracks in the database (library version 1.1)

Field name	Type	Units	Decimals	Optional field
metadata	list of strings	-	-	no
altitude	float	m	1	yes
stops	list of floats	m	1	no
speed limits	list of [float, integer]	[m, km/h]	[1, 0]	no
gradients	list of [float, float]	[m, ‰]	[1, 1]	yes

To fully define the track properties of a specific numerical simulation, some additional parameters may need to be specified. We propose to use a configuration file (also in JSON format) defining upon loading of a track the following fields:

- *id*: the ID of the track to be used in the simulation.
- *from*: index of departure station. 0 if not specified.
- *to*: index of destination station. Last stop if not specified.

3.2 Data Sources

Every track in the benchmark library comes from a scientific publication. We distinguish between the following cases among which the data has become available:

- P for *public*: track information is already publicly available in digital form. Only re-formatting to JSON was necessary.
- D for *described*: the scientific publication contains enough information to accurately reconstruct the track.
- I for *illustrated*: the track is reconstructed only approximately, by digitalizing the figure of a publication with an online tool (WebPlotDigitalizer [13]).

An example of the digitalization process is shown in Fig. 1.

3.3 Content and Structure

The benchmark instances that were collected from the recent literature in TTO are listed in Table 2, together with some of their key properties. TTOBench, which is freely available on GitHub [11], includes both simple benchmark examples as well as real-world track data and it consists of the following files:

- JSON files with track data.
- README files with instructions and references.
- Python files with auxiliary functions (e.g., checking the validity of a track).
- License file (BSD2, which is one of the most permissive licenses).

An interface to import track data to (or export data from) Python has been added to our recently published code in [24], such that the provided optimization framework can be used in combination with TTOBench. We encourage authors in the field to contribute to the benchmark library in form of pull requests to the GitHub repository. The contributions could for example consist of: new tracks, improvement of digitalized tracks or extension of the list of references.

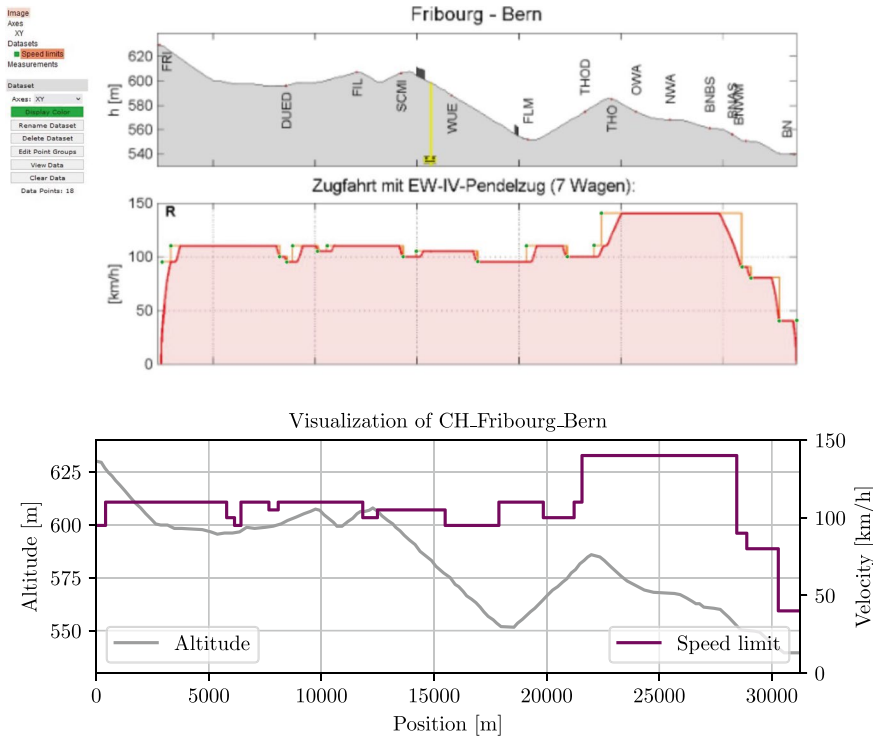


Fig. 1 On the top: original plot from [14] being processed with WebPlotDigitizer. On the bottom: approximate benchmark example imported to database

4 Numerical Simulations

To demonstrate a typical use case of the benchmark library, we compare the performance of three TTO algorithms in terms of energy consumption on every track of the database. The optimization is performed on the complete length of the track, neglecting intermediate stops. The train specifications, adapted from [15], are summarized on Table 3. The constant efficiency factor $\eta = 0.7$ implies that the function of specific power losses in Eq. (1a) is defined as:

$$p^{\text{loss}}(s) = \begin{cases} \frac{1-\eta}{\eta} f^{\text{el}}(s)v(s), & \text{if } f^{\text{el}}(s) \geq 0, \\ \eta f^{\text{el}}(s)v(s), & \text{otherwise.} \end{cases} \quad (3)$$

4.1 Trajectory Calculation

The optimal control problems are discretized with direct multiple shooting and solved with an interior point method [10, 24]. Three problem formulations are considered:

Table 2 Collected benchmark examples in alphabetical order (library version 1.1)

ID	Data	References	Speed limit [km/h]		Gradient [% ϵ]	
			min	max	min	max
00_reference	D	[9, 15]	140	140	0.0	0.0
00_stationX_stationY	P	[10]	80	125	-15.4	15.9
00_var_gradient_minus_5	D	[9]	140	140	-5.0	0.0
00_var_gradient_minus_10	D	[9]	140	140	-10.0	0.0
00_var_gradient_minusplus_6	D	[16]	140	140	-6	6
00_var_gradient_plus_5	D	[9]	140	140	0.0	5.0
00_var_gradient_plus_10	D	[9]	140	140	0.0	10.0
00_var_speed_limit_100	D	[9]	100	140	0.0	0.0
00_var_speed_limit_110	D	[9]	110	140	0.0	0.0
00_var_speed_limit_120	D	[9]	120	140	0.0	0.0
00_var_speed_limit_wind	D	[17]	50	120	0.0	0.0
CH_Stadelhofen_Altstetten	P	[18]	80	125	-38.0	28.0
CH_Fribourg_Bern	I	[14, 19, 20]	40	140	-16.9	14.1
SE_Vasteras_Kolback	I	[6]	110	200	-16.7	10.8
CN_Songjiazhuang_Yizhuang	D	[21–23]	50	84	-24.0	24.0

- **MTTC**: minimum-time train control problem, i.e., train must arrive as fast as possible. The optimal trajectories contain only maximum acceleration, cruising and maximum braking. Yields the highest possible energy consumption. The maximum travel time of the other two methods is equal to the travel time of MTTC plus an additional 15% time reserve.
- **RMS**: reduced maximum speed. With this heuristic method, the maximum allowed speed of the train is iteratively reduced until the MTTC trajectory reaches the maxi-

Table 3 Train specifications NS Intercity VIRM-6 [15]

Property	Unit	Value
Train mass	t	391
Rotating mass supplement	%	6
Maximum speed	km/h	140
Maximum traction force	kN	213.9
Maximum regenerative braking force	kN	142.5
Maximum traction power	kW	2157
Maximum regenerative braking power	kW	3616
Maximum deceleration	m/s ²	0.66
Efficiency factor (traction/regenerative braking)	%	70
Rolling resistance zero-order coefficient	kN	5.8584
Rolling resistance first-order coefficient	kN h/km	0.0206
Rolling resistance second-order coefficient	kN h ² /km ²	0.001

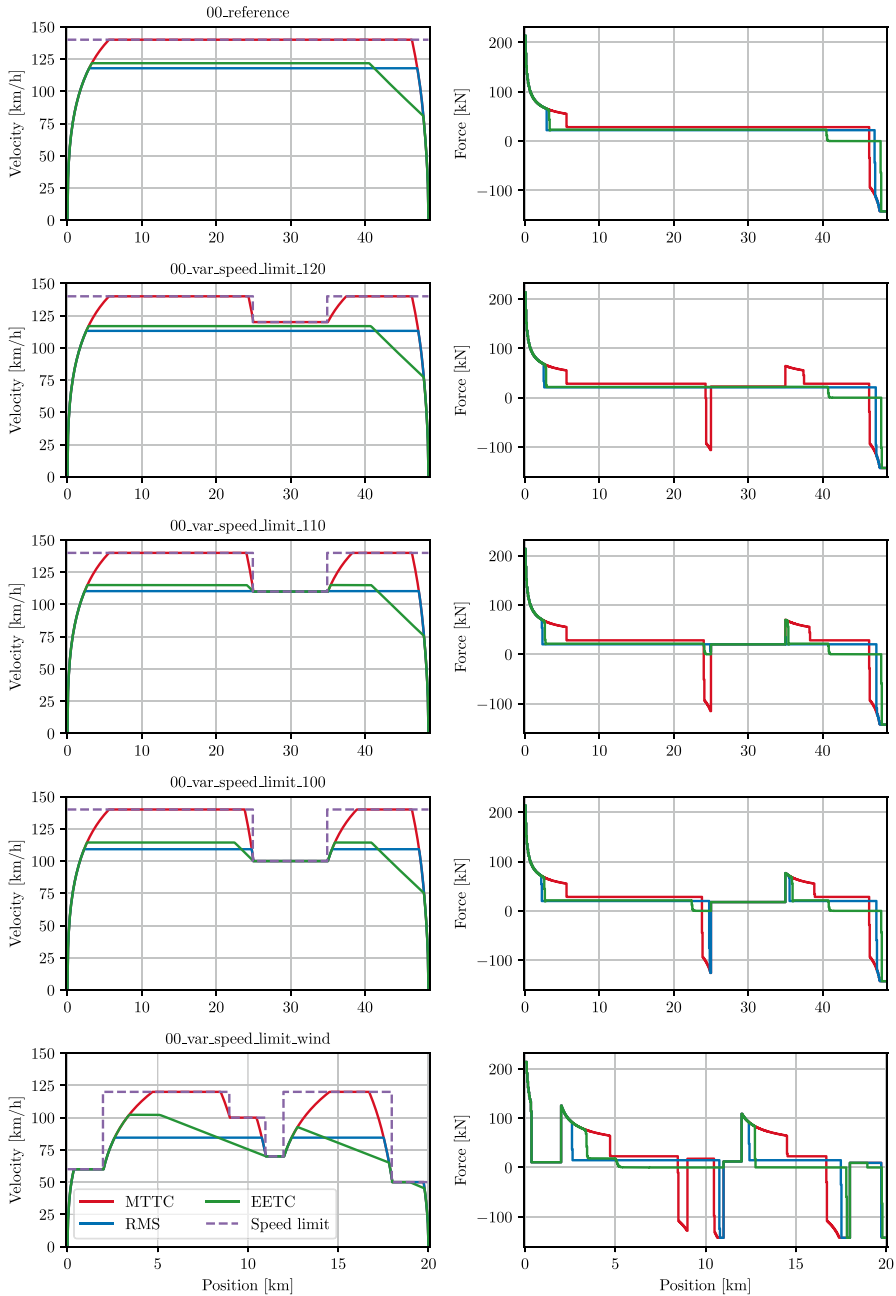


Fig. 2 Time-optimal (red), reduced-maximum-speed (blue) and energy-optimal (green) trajectories on benchmark problems with varying speed limits (purple) on level track

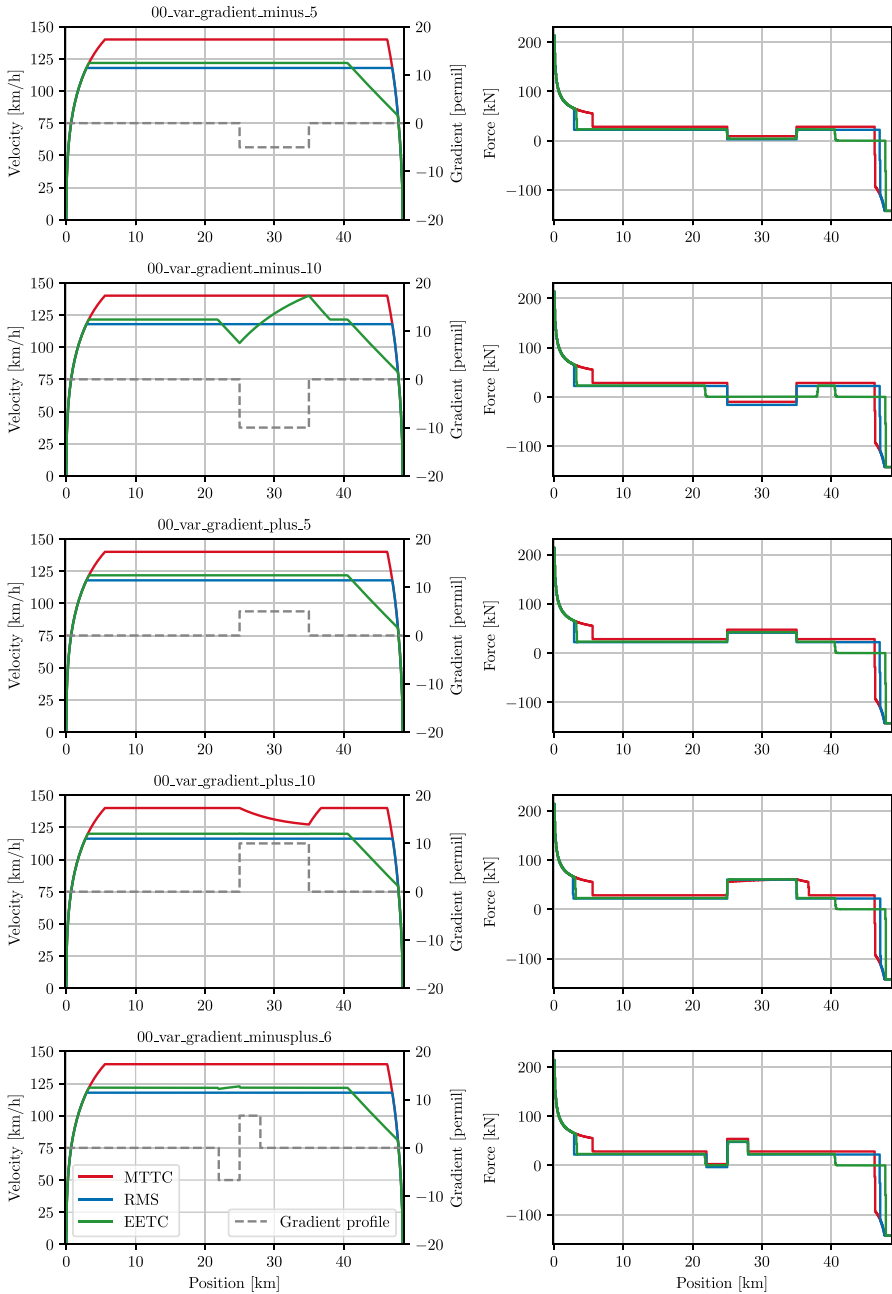


Fig. 3 Time-optimal (red), reduced-maximum-speed (blue) and energy-optimal (green) trajectories on benchmark problems with varying gradients (gray) and constant speed limit

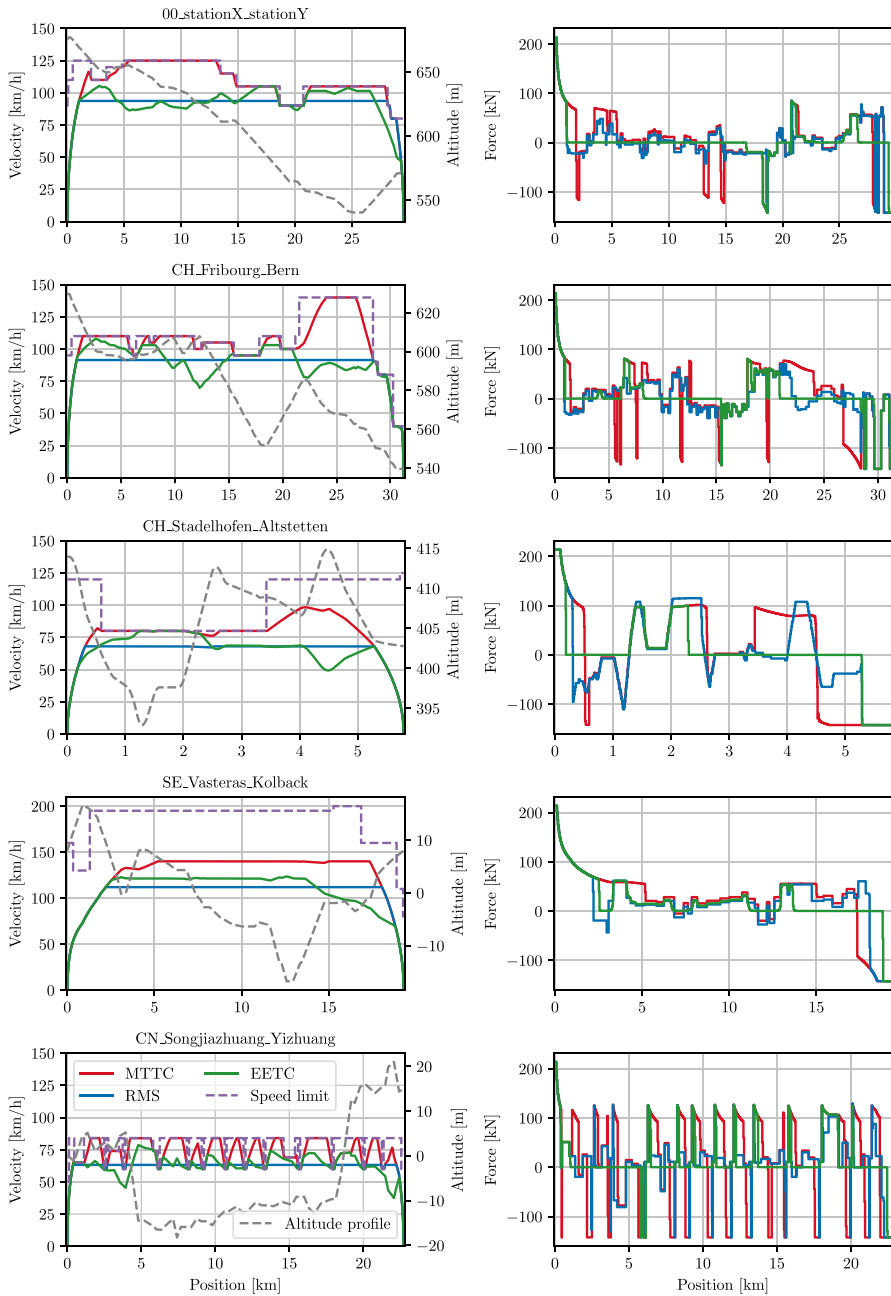


Fig. 4 Time-optimal (red), reduced-maximum-speed (blue) and energy-optimal (green) trajectories on benchmark problems with varying speed limits (purple) and gradients (altitude profile in gray)

Table 4 Travel time and energy consumption of intercity train on different tracks

Track ID	Figure	Travel time [s]		Energy consumption [kWh]		
		MTTC	EETC	RMS		
				MTTC	RMS	EETC
00_reference	2	1370	1576	574.8	456.0	440.5
00_var_speed_limit_120	2	1418	1632	562.2	432.5	417.7
00_var_speed_limit_110	2	1452	1670	555.8	418.1	404.0
00_var_speed_limit_100	2	1492	1716	549.3	409.3	390.3
00_var_speed_limit_wind	2	872	1003	206.3	134.8	120.6
00_var_gradient_minus_5	3	1370	1576	498.7	379.9	364.4
00_var_gradient_minus_10	3	1370	1576	442.8	336.6	290.6
00_var_gradient_plus_5	3	1370	1576	650.9	532.1	516.6
00_var_gradient_plus_10	3	1388	1596	716.4	599.8	584.3
00_var_gradient_minusplus_6	3	1370	1576	574.7	458.1	440.5
00_stationX_stationY	4	1054	1213	166.7	103.4	62.6
CH_Fribourg_Bern	4	1164	1339	228.8	137.0	100.3
CH_Stadelhofen_Altstetten	4	304	351	58.9	41.5	24.3
SE_Vasteras_Kolback	4	619	713	245.7	193.3	167.7
CN_Songjiazhuang_Yizhuang	4	1167	1343	287.0	170.7	138.3

imum allowed travel time (with a tolerance of 0.5 s). The optimal control regimes are again maximum acceleration, cruising and maximum braking.

- EETC: energy-efficient train control problem, i.e., train must arrive using the least possible energy. Under the assumption of constant efficiency, the optimal trajectories consist of maximum acceleration, cruising, coasting and maximum braking, arriving at the destination as late as possible.

4.2 Benchmark Results

The optimal speed profiles and respective electrical force at the wheel are shown in Figs. 2, 3, and 4, while the results are summarized on Table 4.

Figure 2 contains trajectories on level track with varying speed limits. The RMS algorithm reduces the total energy consumption by 26% on average with respect to the worst-case scenario (MTTC), while the EETC trajectories bring an additional 5% improvement.

Figure 3 depicts the trajectories on the tracks with constant speed limit and varying gradients. Here, the situation is fairly similar: the RMS algorithm reduces the energy consumption by 21% on average and the EETC algorithm by another 5%.

Based on the simple benchmark examples of Figs. 2 and 3, one could draw the conclusion that the additional benefit of EETC with respect to the RMS method is rather limited. However, the results change significantly once realistic tracks with varying speed limits and gradients are considered. According to Table 4, the RMS

trajectories in Fig. 4 reduce the total energy consumption by 23%, yet the EETC algorithm brings an additional improvement of 28%.

5 Conclusion

This paper presents an effort to improve the data availability in the field of TTO, since we believe that repeatability of the results of publications in the field is an essential step for the derivation of better algorithms and software.

As a first step towards a comprehensive benchmark library, we provide a database of track infrastructure data including gradients and speed limits. We envision that the content of TTOBench will grow further than that, steered by the interest and participation of the community. In the list below, we summarize the content of the library at the time of writing, a tentative collection of future features and what is currently considered out of scope:

- *Current status*: version 1.1 of TTOBench provides a detailed description of track-related information in JSON. 15 tracks have been collected from the literature and brought into this format to enable easier benchmarking of different TTO algorithms.
- *Future extensions*: the structure of the JSON files has been carefully designed to allow for straightforward extensions in the near future. Trains shall be also described by means of comprehensive JSON files, a library of trains used in scientific papers or published by train manufacturers shall be collected and the track description shall be populated with more fields (e.g., curvature and tunnel information). Finally, other relevant information, such as timetable and weather data, shall be additionally described in JSON to complete the definition of elements that may play a role in TTO.
- *Out of scope*: although we recognize the relevance of interconnections between tracks and interlocking information in the field of TTO, we consider these categories to be out of the current scope of the library.

Appendix. JSON File Example

An example of a JSON file describing a 10-km-long track with one speed limit change and one gradient change is shown in Listing 1. The speed limit is decreased from 140 km/h to 100 km/h in the middle of the track and a positive (i.e., uphill) slope of 10‰ starts 2 km before the end of the track.

```
{
  "metadata": {
    "id": "OO_example",
    "created by": "Dimitris Kouzoupis",
    "library version": "TTOBench v1.1"
  },
  "stops": {
    "unit": "m",
    "values": [
      0.0,
      10000.0
    ]
  },
  "speed limits": {
    "units": {
      "position": "m",
      "velocity": "km/h"
    },
    "values": [
      [
        0.0,
        140
      ],
      [
        5000.0,
        100
      ]
    ]
  },
  "gradients": {
    "units": {
      "position": "m",
      "slope": "permil"
    },
    "values": [
      [
        0.0,
        0.0
      ],
      [
        8000.0,
        10.0
      ]
    ]
  }
}
```

Listing 1 JSON file example of a simple track

Author Contribution Dimitris Kouzoupis wrote the manuscript text and collected the data for the benchmark library. Ishan Pendharkar and Francesco Corman contributed with ideas and reviewed the manuscript.

Funding Open access funding provided by Swiss Federal Institute of Technology Zurich This research was supported by NCCR Automation, with grant number 51NF40_180545.

Data Availability The data sets analysed during the current study are available in the GitHub repository: *Open-source library for train trajectory optimization* under the link <https://github.com/dkouzoup/TTOBench>.

Code Availability The code of the case study is not publicly available due to little relevance to the contributions of the paper but is available from the corresponding author on reasonable request.

Declarations

Ethics Approval Not applicable.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

Conflict of Interest Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Scheepmaker GM, Goverde RMP, Kroon LG (2017) Review of energy-efficient train control and timetabling. *Eur J Oper Res* 257:355–376
2. Strobel H, Horn P, Kosemund M (1974) Contribution to optimum computer-aided control of train operation. In: 2nd IFAC/IFIP/IFORS Symposium on Traffic Control and Transportation Systems, pp 377–387
3. Howlett PG, Pudney PJ (1995) *Energy-efficient train control*. Springer, London
4. Franke R, Terwiesch P, Meyer M (2000) An algorithm for the optimal control of the driving of trains. In: Proceedings of the IEEE Conference on Decision and Control (CDC)
5. Nold M, Corman F (2022) Modelling realistic energy losses from variable efficiency and vehicle systems. In: Determining Energy Efficient Train Control. Swiss Transport Research Conference (STRC)
6. 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. *Transport Res C Emerg Technol* 85:111–130
7. Wang Y, Schutter BD, van den Boom TJJ, Ning B (2013) Optimal trajectory planning for trains - a pseudospectral method and a mixed integer linear programming approach. *Transport Res C Emerg Technol* 29:97–114
8. Wang P, Goverde RMP (2016) Multiple-phase train trajectory optimization with signalling and operational constraints. *Transport Res C Emerg Technol* 69:255–275
9. Goverde RMP, Scheepmaker GM, Wang P (2021) Pseudospectral optimal train control. *Eur J Operat Res* 292:353–375
10. Kouzoupis D, Pendharkar I, Frey J, Diehl M, Corman F (2023) Direct multiple shooting for computationally efficient train trajectory optimization. *Transport Res C Emerg Technol* 152

11. Kouzoupis D (2022) GitHub repository: Public library of train trajectory optimization algorithms. <https://github.com/dkouzoup/TTOBench>
12. Bärman A, Gemander P, Hager L, Nöth F, Schneider O (2023) EETLib-energy-efficient train timetabling library. *Networks* 81(1):51–74
13. Rohatgi A (2022) WebPlotDigitizer. <https://automeris.io/WebPlotDigitizer>
14. Meyer M, Menth S, Lerjen M (2007) Potentialemittlung Energieeffizienz Traktion bei den SBB. Technical report, Bundesamt für Energie BFE
15. Scheepmaker GM, Goverde RMP (2020) Energy-efficient train control using nonlinear bounded regenerative braking. *Transport Res C Emerg Technol* 121:102852
16. Scheepmaker GM, Willeboordse HY, Hoogenraad JH, Luijt RS, Goverde RMP (2020) Comparing train driving strategies on multiple key performance indicators. *J Rail Trans Plan Manage* 13:100163
17. Trivella A, Wang P, Corman F (2021) The impact of wind on energy-efficient train control. *EURO J Transport Logist* 10:100013
18. Mäder D (2023) Entwicklung eines tools für die auslegung von traktionsbatterien für die netzunterstützung. Master's thesis, FHNW
19. Aradi S, Bécsi T, Gáspár P (2013) A predictive optimization method for energy-optimal speed profile generation for trains. In: *IEEE International Symposium on Computational Intelligence and Informatics*, pp 135–139
20. Aradi S, Bécsi T, Gáspár P (2014) Design of predictive optimization method for energy-efficient operation of trains. In: *Proceedings of the European Control Conference (ECC)*, pp 2490–2495
21. Yang X, Li X, Ning B, Tang T (2015) An optimisation method for train scheduling with minimum energy consumption and travel time in metro rail systems. *Transp B Trans Dyn* 3(2):79–98
22. Ye H, Liu R (2016) A multiphase optimal control method for multi-train control and scheduling on railway lines. *Transport Res B* 93:377–393
23. Zhong W, Lin Q, Loxton R, Teo KL (2021) Optimal train control via switched system dynamic optimization. *Optim Methods Softw* 36(2–3):602–626
24. Kouzoupis D (2022) GitHub repository: Direct multiple shooting for energy efficient train control. <https://github.com/dkouzoup/ms-eetc>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.