# A data-driven bi-objective matheuristic for energy-optimising timetables in a passenger railway network 

Matthias Villads Hinsch Als *, Mathias Bejlegaard Madsen, Rune Møller Jensen<br>IT University of Copenhagen, Rued Langgaards Vej 7, 2300 Copenhagen S, Denmark

## A R T I CLE I N F O

## Keywords:

Multi-objective optimisation
Energy-efficient train timetabling
Matheuristic
Mixed-integer programming
Genetic algorithm
Decision support


#### Abstract

Energy-efficient train timetabling (EETT) is essential to achieve the full potential of energyefficient train control, which can reduce operating costs and contribute to a reduction in $\mathrm{CO}_{2}$ emissions. This article proposes a bi-objective matheuristic to address the EETT problem for a railway network. To our knowledge, this article is the first to suggest using historical data from train operation to model the actual energy consumption, reflecting the different driving behaviours. The matheuristic employs a genetic algorithm (GA) based on NSGA-II. The GA uses a warm-start method to generate the initial population based on a mixed-integer program. A greedy first-come-first-served fail-fast repair heuristic is used to ensure feasibility throughout the evolution of the population. The objectives taken into account are energy consumption and passenger travel time. The matheuristic was applied to a real-world case from a large North European train operating company. The considered network consists of 107 stations and junctions, and 18 periodic timetables for 9 train lines. Our results show that for an entire network, a reduction up to $3.3 \%$ in energy consumption and $4.64 \%$ in passenger travel time can be achieved. The results are computed in less than a minute, making the approach suitable for integration with a decision support tool.


## 1. Introduction

Railway transportation becomes an increasingly important mode of transportation to meet future travel demands. Especially the energy efficiency of railway transportation compared to air and other ways of land travel becomes vital in the global attempt to reduce emissions of $\mathrm{CO}_{2}$ and other greenhouse gasses to curb climate change. For this reason, the EU declared 2021 to be the year of rail (Keersmaecker and Meder, 2020) to promote sustainable travel and to work towards the goals of making Europe the first climate-neutral continent by 2050 and cutting $55 \%$ of the $\mathrm{CO}_{2}$ emissions before 2030 (European Commission, 2019). Additionally, a reduction in energy consumption is desirable by train operating companies (TOC) to reduce costs. TOC have taken many initiatives to cut $\mathrm{CO}_{2}$ emissions, but optimising the operation of the trains is one way to achieve immediate energy-efficiency improvements. One crucial element to utilise the full energy-saving potential is to look into energy-optimisation of the timetable planning process to reduce energy consumption.

The train timetabling problem (TTP) is generally known to be an NP-hard problem (Higgins et al., 1997; Caprara et al., 2002; Garrisi and Cervelló-Pastor, 2019). For that reason, in practice, timetable planners only make minor changes to ensure a feasible timetable fulfilling contractual agreements and business key performance indicators (KPIs). Timetable planners seldom have insight

[^0]into the energy efficiency of the produced timetables. Therefore, research is needed to assist timetable planners in finding feasible energy efficient timetables.

Common for most previous work on TTP is that their proposed methods do not take the drivers' actual train control into account, hence not reflecting real-world conditions. Fortunately, many TOCs today use driver advisory systems (DAS) that are capable of collecting data such as position, speed, and arrival/departure times. This valuable data can be used to learn the actual driving behaviour to better model the actual energy consumption. To this end, we have collaborated with Cubris (Cubris - A. Thales Company, 2021) that delivers a DAS called GreenSpeed to European TOCs. As a part of improving the provided service to their customers, more than 140,000 regional train runs were available to the authors of this article to learn the relations between actual running times and energy consumption. The usage of this kind of historical data for energy-optimising a network of timetables will be a new contribution to this field of research. To our knowledge, only Madsen et al. (2019) use this kind of data to estimate energy consumption in EETT and obtain promising results by saving $33.07 \%$ energy for a single section and $6.23 \%$ for a timetable. However, the problem considered in their paper is only for a single timetable with a single objective.

In addition to that, this article will contribute to research by proposing a matheuristic to optimise the energy consumption of a large railway network using operational data. The proposed matheuristic is a hybrid between a mixed-integer program and a genetic algorithm (GA), where the initial population for the GA is generated by the mixed-integer program. This article will consider the multi-objective network-optimised EETT (MONO-EETT) problem. To the best of our knowledge, no attempts have been made yet at using historical data from the train operation of an entire railway network in relation to EETT for multiple objectives.

The approach presented in this article builds on top of the work of Madsen et al. (2019) showing how historical data could be used for learning the relationship between actual run times and energy consumption, referred to as the energy functions or trade-off curves. These energy functions can be used for energy-optimising the timetable. This article extends that approach by using the learnt energy functions to optimise an entire railway network with multiple lines using a matheuristic. The proposed matheuristic uses a GA based on NSGA-II (Deb et al., 2002) and a warm-start method to generate the initial population using a commercial MIP solver. We call this a MIP-initialiser. The two objectives considered are energy consumption and passenger travel time. The GA uses a crossover that is a hybrid between the average and single-point crossovers. Each individual is mutated according to a Gaussian distribution after the crossover and then repaired to ensure no constraints are violated. The repair heuristic is greedy, first-come-first-serve (FCFS), and fail-fast to keep the repair time at a minimum.

The proposed approach was evaluated on a real-world case study with historical data from a large North European TOC. The case study contains 18 timetables on a network with 107 stations and junctions. The matheuristic showed promising results with energy savings up to $3.3 \%$ and $4.64 \%$ reduction in passenger travel time without sacrificing each other. If significant trade-offs between the two objectives are allowed, the TOC can save up to $4.83 \%$ energy and $8.98 \%$ passenger travel time. In the worst case, the matheuristic spent less than a minute finding the set of Pareto-optimal solutions. The short computation time makes the solution approach suitable to be used in decision support to guide the timetable planners in making better-informed decisions.

The remaining part of the article is structured as follows. In Section 2, the relevant and recent work will be presented. In Section 3 we will formulate the problem considered by this article, and Section 4 will in detail describe the proposed algorithm. The experiments and results will be discussed in Section 5, and finally, conclusions will be drawn, and future work will be presented in Section 6.

## 2. Related work

The train timetabling problem (TTP) has attracted an extensive amount of contributions to various formulations of the problem. Most contributions have been reviewed in several comprehensive surveys which also defines the efficient modelling of the problem (Assad, 1980, 1981; Haghani, 1987; Cordeau et al., 1998; Huisman et al., 2005; Caprara et al., 2007; Lusby et al., 2011; Cacchiani and Toth, 2012; Harrod, 2012; Caimi et al., 2017; Lusby et al., 2018). In the literature, the formulations distinguish between periodic and aperiodic timetables, also referred to as cyclic and non-cyclic timetables. Among these two formulations, the problem described is based on either Integer Programming (IP) or Mixed-Integer Programming (MIP). In recent literature, the decision variables of time have been made continuous (i.e. the arrival and departure times) while keeping the ordering of the trains discrete (Caprara et al., 2007). For this reason, MIP has been extensively applied. In the literature regarding periodic timetables, most papers represent the problem based on the Periodic Event Scheduling Problem (PESP) by Serafini and Ukovich (1989) formulated as an IP. In PESP, an event represents the arrival at or the departure from a given station. The event is scheduled for one cycle in such a way that the cycle can be repeated. However, the PESP formulation by Serafini and Ukovich (1989) assumes that the trip times of the trains between consecutive stations are known and fixed a priori. Kroon and Peeters (2003) therefore extended PESP with variable trip times to overcome this.

TTP is generally known to be an NP-hard problem (Higgins et al., 1997; Caprara et al., 2002; Garrisi and Cervelló-Pastor, 2019). This makes traditional exact methods to solve IPs and MIPs computationally too time-consuming when the size and the complexity of the problem gets big. In these cases, approximate solutions found by heuristic approaches are preferred. Table 1 summarises the relevant contributions for the problem defined in this article which will be outlined throughout this literature review. The papers have been selected based on being multi-objective or considering a large network instance, in order to highlight the common approaches used.

For many years the common objective for TTP has been to either minimise the passenger travel time, minimise the total accumulated delay, or minimise the number of conflicts (Assad, 1980, 1981; Haghani, 1987; Cordeau et al., 1998; Huisman et al., 2005; Caprara et al., 2007; Lusby et al., 2011; Cacchiani and Toth, 2012; Harrod, 2012; Caimi et al., 2017; Lusby et al., 2018). In

Table 1
Related work summary table.

| Reference | Formulation |  | \#Objectives |  | Infrastructure |  | Approach |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | IP | MIP | Single | Multi | Line | Network |  |
| Yang et al. (2019) | $\times$ |  |  | $\times$ | $\times$ |  | GA (NSGA-II). |
| Xu et al. (2016) | $\times$ |  |  | $\times$ | $\times$ |  | GA. |
| Semet and Schoenauer (2005) |  | $\times$ | $\times$ |  |  | $\times$ | Matheuristic (GA-MIP). |
| Wang et al. (2019) |  | $\times$ |  | $\times$ | $\times$ |  | GA-PSO. |
| Tormos et al. (2008) |  | $\times$ |  | $\times$ | $\times$ |  | GA. |
| Higgins et al. (1997) |  | $\times$ | $\times$ |  | $\times$ |  | LS, TS, GA, GA-LS, GA-TS. |
| Nachtigall and Voget (1997) |  | $\times$ |  | $\times$ |  | $\times$ | Fuzzy GA. |
| Madsen et al. (2019) |  | $\times$ | $\times$ |  | $\times$ |  | MIP. |
| This article |  | $\times$ |  | $\times$ |  | $\times$ | Matheuristic (GA-MIP). |

recent years there has been a shift towards minimising the total energy consumption due to the benefits of a significant reduction in $\mathrm{CO}_{2}$ emissions and energy consumption. However, the research on EETT which redistributes slack time to maximise the potential of energy-efficient train control (EETC) is still limited (Scheepmaker et al., 2020). The existing contributions to EETT and EETC have been reviewed by Yang et al. (2016) and Scheepmaker et al. (2017). Scheepmaker et al. (2017) describes the relationship between slack and energy as when the slack increases then less energy is needed. This is due to the train being able to run at a lower cruising speed or starting earlier with coasting. If not considering regenerative braking, then the trade-off between running time and energy consumption represents the multi-objective function, also referred to as the trade-off curve. Common for the papers mentioned in Scheepmaker et al. (2017), EETT is solved as a two-step approach, where the first step is estimating the trade-off curve which is then used in the second step to optimise the timetables. Generally, the trade-off curve is estimated per trip based on simulations where an EETC problem is solved iteratively for each running time instance. Some papers also included uncertainties of the performance of the train driving and the delays by adding fuzzy variables to the simulations.

Within EETT, there exist different branches of research. One branch takes regenerative braking into account to synchronise regenerative braking trains arriving at a station with departing accelerating trains in order to minimise loss of regenerated energy (Scheepmaker et al., 2017). One of the latest contributions in this area is by Yang et al. (2015). They propose a genetic algorithm (GA) to maximise the utilisation of regenerative energy capable of decreasing energy consumption by $6.97 \%$. The formulation considered in this article does not take regenerative braking into account due to the type of trains used in this article does not have regenerative brakes.

A different branch with a similar formulation to the one considered in this article is studying the optimal distribution of slack time for trains over multiple stops in a single timetable. Sicre et al. (2010) use Pareto optimisation to redistribute slack time among the sections in a journey achieving a decrease of $33.63 \%$ in energy consumption compared to the energy consumption of the maximum speed profile. Cucala et al. (2012) propose a fuzzy linear programming model taking the behavioural response of the driver into account. The approach achieved a decrease of $6.7 \%$ in energy consumption compared to the timetable in service. However, both Sicre et al. (2010) and Cucala et al. (2012) model the energy consumption using simulated data not reflecting the actual driving behaviour and real-world conditions. Another branch of research takes instead EETC into account when finding the optimal distribution of slack time. Su et al. $(2013,2014)$ propose an analytical algorithm to obtain energy-efficient timetables by calculating an energyefficient speed profile for the whole journey based on the Pontryagin maximum principle. The proposed algorithm reduces the energy consumption by $10.3 \%$ on average for a single interstation and $14.5 \%$ for the entire journey. Though naturally, EETC does not either reflect the actual train control of the drivers and real-world conditions.

Common for the aforementioned papers is that they only take the single objective of minimising energy consumption into account. A popular choice when dealing with multi-objective optimisation problems is evolutionary algorithms such as GAs. To our knowledge, the first to suggest a GA approach for TTP was Nachtigall and Voget (1997). The paper formulates a bi-objective problem of minimising the waiting time and costs in integrated fixed interval timetables. The formulation is based on PESP, formulated as a MIP, and is solved by a fuzzy GA incorporating a greedy starting method and a local improvement algorithm. Arenas et al. (2015) propose a GA with near-optimal results achieved within short computation times. In this formulation, a macroscopic representation takes both infrastructure and rolling stock elements into account to optimise the train journey scheduling based on the global profit generated. Garrisi and Cervelló-Pastor (2019) explored the possibilities of applying parallel genetic metaheuristics with real-time concurrency to train scheduling. The paper proposes a GA to find an approximate solution to a multi-objective problem. To improve their results, they use heuristic techniques to generate an initial population. The approach produces feasible solutions in seconds. The paper compares the performance of their proposal against the results of Arenas et al. in Arenas et al. (2015) which shows that for a large number of trains, their solution obtains the optimal train scheduling faster. Wang et al. (2019) propose a hybrid method between a GA and particle swarm optimisation (PSO) for the timetable rescheduling problem. The objectives are to minimise the total delays of all involved trains and the number of trains delayed. The paper formulates a macroscopic model, where stations are regarded as nodes with a given capacity, which is the number of siding tracks. The solution was tested on the Beijing-Shanghai high-speed railway corridor. Their results show that the proposed method found the best solution within 1.5 min .

Xu et al. (2016) formulate a multi-objective optimisation problem of minimising passenger travel time and energy consumption using a GA. However, the multi-objective model is transformed into a single-objective model before it is given to the GA. They are able to decrease the energy consumption by at most $9.96 \%$ and the passenger travel time by $7.51 \%$ compared to the current
timetable. Yang et al. (2019) use NSGA-II to optimise timetables in the Beijing Metro with respect to three objectives: cost, passenger waiting time, and robustness. They achieve a reduction in energy consumption by $42.1 \%$, a reduction in passenger waiting time by $15.8 \%$, and increased robustness by $24.81 \%$. Yin et al. (2017) propose a Lagrangian relaxation-based heuristic to reduce passenger waiting time and energy consumption of the Beijing Metro. Using real-world passenger demand data, the computational results showed that the heuristic was able to achieve satisfactory solutions in a short computational time compared to using a CPLEX solver. Huang et al. (2017) propose a Tabu Search algorithm to minimise the multi-objective optimisation problem of passenger total travel time and energy consumption. Numerical experiments were conducted on the Yizhuang Line in Beijing. Gao et al. (2018) formulate a bi-objective linear program, which considers energy consumption and passenger travel time. They showed that the energy consumption was reduced by $9.4 \%$ and passenger travel time was reduced by $5.4 \%$ for the Beijing Metro Line 6 . Tang et al. (2021) propose an adaptive GA (A-GA) to reduce passenger travel time that is taking train capacity, overtaking, and other operational constraints into account. The problem was formulated as a mixed-integer nonlinear program (MINLP). Their results show that the A-GA was able to obtain near-optimal solutions with significantly less computation time than solving the MINLP using a commercial solver.

Semet and Schoenauer (2005) present a matheuristic combining a permutation-based evolutionary algorithm with the commercial MIP solver, CPLEX. The evolutionary algorithm uses a semi-greedy heuristic to gradually reconstruct the schedule by inserting trains one after another following the permutation. CPLEX is employed after the evolutionary algorithm to refine the solution. Their results show that their approach obtains better and faster results for a large real-world case, however less efficient on average.

Other relevant papers are, for instance, Higgins et al. (1997) who study the single-line train scheduling problem with a single objective and compare six different heuristics: a local search heuristic with an improved neighbourhood structure, GA, tabu search, and two memetic algorithms. The two memetic algorithms are based on local search and tabu search respectively. Their results show that both memetic algorithms perform better than the others but with an increased computation time up to seven times that of the GA. The GA and the two memetic algorithms give solutions within $5 \%$ of the optimal for $90 \%$ of the problems tested. It finds the optimal solution for nearly $50 \%$ of these. Tormos et al. (2008) propose a GA for solving the TTP formulated as a job-shop scheduling problem for multiple objectives. The paper represents a solution as an activity list of pairs consisting of a train and a track section of its journey which represents a job. The paper tested its approach on real instances obtained from the Spanish Manager of Railway Infrastructure (ADIF) which yielded satisfactory results within seconds.

Madsen et al. (2019) propose a five-step approach to TTP using historical data from train operation to predict the energy consumption of trains as a function of section run time using clustering and regression. The learnt relations are used in a mathematical program and displayed in a decision support tool. To our knowledge, they are the first to propose using operational data from rolling stock to estimate energy consumption in EETT. Their results are promising by saving $33.07 \%$ energy for a single section and $6.23 \%$ for a whole timetable. However, the problem considered in their paper is only for a single timetable with a single objective.

Through the literature review, it has become apparent that further research on EETT with multiple objectives is still necessary. For the papers that do multi-objective optimisation, it is relevant to look further into matheuristic approaches in order to achieve better Pareto optimal solutions. Furthermore, only a few papers consider more than a single timetable for a large railway network. Last, to the best of our knowledge, we are the first to use extensive historical data from the train operation of an entire railway network to model actual trade-off curves between energy consumption and running time in TTP. With this approach, the actual speed profiles of the train drivers are modelled, as an alternative to using optimal speed profiles from EETC when simulating the trade-off curves.

## 3. Problem formulation

This section will introduce the train timetabling problem (TTP) domain and describe the version of TTP considered in this article. Furthermore, the corresponding mathematical formulation and their underlying assumptions of the railway network will be presented.

### 3.1. Network infrastructure model

A railway network consists of a set of stations that are connected by tracks. A track connecting two stations is called a section. The network can be viewed from different levels of detail, which is usually characterised as microscopic or macroscopic networks. The difference is visualised in Fig. 1. Microscopic networks carry the greatest detail about the infrastructure and usually contain details about how tracks are connected by switches and where signals are placed on sections. On the other hand, macroscopic networks only describe how stations are interconnected by sections and with what capacity. A macroscopic network can be represented as a graph with nodes representing stations and junctions and edges representing sections. Junctions can, in many cases, be regarded as "virtual stations" without platforms and do not allow stopping. While microscopic networks are used to generate high-quality simulations, that level of detail is often not needed or available during long-term planning which is the focus of this article. Therefore, similar to most other research in TTP (Arenas et al., 2015; Tormos et al., 2008; Wang et al., 2019; Semet and Schoenauer, 2005; Yang et al., 2019; Xu et al., 2016; Madsen et al., 2019), this article will continue with a macroscopic network model, however, with a few modifications.

In order to know the accessible capacity at each station, this article introduces the concept of platform groups to the network model. A platform group is a set of platforms at a station with similar properties. Platform groups are distinguished by reachable

## Microscopic network model



## Macroscopic network model



## Our network model



Fig. 1. Differences in microscopic, macroscopic, and our infrastructure models. The capacities are given for one direction due to the symmetry assumption.
stations for downstream stops in the schedule, and if the platform group allows for stopping. Each platform group has a capacity indicating how many trains can pass through or dwell (if allowed) at the same time. This concept becomes handy in two cases: (1) Some destinations may not be accessible from all tracks or platforms at a station or junction. Therefore, different destinations may need stopping at different platform groups at a station resulting in different capacities. (2) Many stations have several tracks, but only some tracks have platforms. Thus, trains can pass through at the station while another train dwells, which yields a higher capacity for trains passing through. These cases are shown in Fig. 2 and can be seen in a network context in Fig. 1. In our model, there is no difference between stations and junctions. Rather, junctions are a special case of stations that also has platform groups, but the train cannot stop at any of the platform groups. As stops will not be moved, inserted, or removed in the optimised timetable, this missing distinction causes no problems. Based on the macroscopic formulation, this article makes a number of simplifying assumptions:

- A track can only have unidirectional traffic. Though, a section can have multiple tracks. This assumption does not support single-track sections.
- Sections and stations are symmetric in the sense that if two parallel tracks are going in one direction, the same number of tracks is going in the opposite direction. Analogously, stations have the same number of platforms for trains going in both directions. Turnaround platforms, which are platforms where a train can only enter and exit from one direction, are an exception.
- The network forms an undirected acyclic graph, in essence a linear network with no loops. An example of a linear network is provided in Fig. 3. The notation will be introduced later.

These assumptions reduce the number of applications for which the solution approach can provide feasible solutions. However, scenarios not supported by the assumptions can still be modelled as closely as possible to reality and let the timetable planner make the final adjustments to make the timetable feasible. In these cases, even near-feasible timetables are also helpful to the timetable planner.

### 3.2. The train timetabling problem

The version of TTP treated in this article considers timetables which schedule a given train on a train line in a railway network. The train line specifies the origin and destination station along a predetermined fixed route. The timetable specifies the arrival and departure times for each visited station on the route. The time it takes to run a section is called the section run time, which has minimum and maximum time constraints. Similarly, the amount of time a train is stopped at a station, called dwell time, also has minimum and maximum time constraints. The difference between the section run time and the minimum run time is called the slack time which can be adjusted in order to optimise the timetable for energy consumption or other objectives. A timetable is valid for exactly one journey, i.e. exactly one departure from the start station unless it is periodic where the exact timetable is repeated with a given cycle time.


Fig. 2. Two cases where platform groups leads to more fine-grained control over station capacity.

Train $i$


Fig. 3. Visual representation of network notation.

Passenger trains in Europe are cyclic for the convenience of the passengers (Caprara et al., 2007; Harrod, 2012), giving rise to the periodic TTP (PTTP). In practice, trains enter and leave service at different times of the day with different cycle times and number of repetitions. It requires a significant amount of computational effort to ensure that all train services' cycles are conflict-free in a railway network where many train lines intersect. Therefore, this article approach PTTP by modelling the busiest hour of the day, effectively eliminating the periodic complexity and leaving us with a standard TTP. The timetable planner can then easily repeat the generated timetables as needed and take trains out in the hours not needed without introducing new conflicts.

The timetable planning process is complex and time-consuming and is largely dominated by manual work by the timetable planner. Planning a timetable usually involves multiple reiterable steps, some of them are (Arenas et al., 2015; Lusby et al., 2011):

1. Line planning: Determine routes and frequency of service according to agreements and contracts.
2. Timetable planning: Plan actual timetables according to infrastructure constraints, often based on an existing set of timetables.
3. Crew and rolling stock planning: Select specific locomotives and wagon units to run the timetable and schedule crew to operate them.
4. Real-time planning: Recover timetables and minimise delays after disruption of operation.

During one of the steps, the timetable planner may find out that the plan is infeasible and need to reiterate one or more of the previous steps to add or remove trains or stops to make the plan feasible. For this reason, more significant changes are typically avoided because of the high combinatorial complexity caused by the numerous infrastructure constraints and KPIs to consider.

The timetable planner possesses important domain knowledge that reaches beyond these constraints. Some knowledge has been passed down through many generations of planners and is difficult to model without significantly increasing the complexity of the problem being solved. Therefore, the proposed solution in this article is not meant to replace the timetable planner but rather to


Fig. 4. Optimal speed profiles and cumulative energy consumption.
assist them in making well-informed decisions, for example, through a decision support tool. Our focus is the timetable planning step, considering that the solution should fit into their daily work. Therefore, our version of the TTP will allow for locking individual arrival/departure times. There are two reasons for this: (1) it allows the timetable planner to use their domain knowledge to lock times through the decision support tool they know are inappropriate to optimise, for example, at busy central stations, and (2) it allows us to insert timetables into the optimisation without optimising them. In practice, the timetable planners will only optimise a subset of timetables to avoid significant changes. The timetables not in the subset will be included as constraints in the optimisation.

The version of TTP considered in this article is assumed to already have a set of existing timetables for different train lines on a regional railway network. Thus, each timetable also has a fixed route through the network. The algorithm proposed in this article will then output a new set of timetables by only adjusting the section run times and station dwell times. Changing the route of the train journey, station platform assignment, and addition/removal of stops is beyond the scope of this article.

### 3.3. Energy consumption estimation

According to literature (Su et al., 2013; Scheepmaker and Goverde, 2015) the potential of energy-efficient train control can be uniquely determined by the amount of running time in the timetable, which in the end determine the amount of energy saved. Eq. (1) describes the kinetic energy associated with the state of motion of an object assuming a constant force:

$$
\begin{equation*}
E_{K}=\frac{1}{2} m v^{2}=\frac{m \Delta s^{2}}{2 \Delta t^{2}} \tag{1}
\end{equation*}
$$

where $m$ is the mass, $v$ is the velocity, $\Delta s$ is the displacement and $\Delta t$ is change in time. This translates into the work done by the train through the work-kinetic energy theorem, which states that the work done equals the change in kinetic energy. From Eq. (1) it becomes apparent that given a fixed displacement, then the work decreases quadratically by higher running times, since the necessary speed to reach the next stopping point in time gets lower.

Figs. 4(a) and 4(b) show four optimal speed profiles calculated by the proprietary solver from Cubris. From Fig. 4(b) the difference between the red and green speed profile is noticeable. The red speed profile is the maximum speed profile, where the cruising speed becomes the maximum speed of the train, which it maintains until it starts braking approximately $500-1000 \mathrm{~m}$ before the station. The green speed profile on the other hand selects a lower optimal cruising speed and then switches to the coasting phase earlier than the blue speed profile. Maximising coasting is the best strategy in order to save energy which also becomes apparent in Fig. 4(c). Fig. 4(c) shows the difference in cumulative energy over time between the speed profiles. As initially observed in Eq. (1), the difference in total energy consumption decreases approximately by a quadratic coefficient as observed in Eq. (1). Therefore, it is reasonable to approximate the energy consumption as a function of running time being a second-degree polynomial.

### 3.4. Objectives

Essentially, a timetable should minimise costs and maximise profits for the TOC. These objectives, however, are often not directly measurable in terms of the timetable. While energy consumption is easily converted into cost, passenger travel time (PTT) is not easily converted into profit. PTT is the sum of all passengers' time spent in the train system and is an indicator of the attractiveness of the train service. A high PTT will result in fewer passengers choosing the service resulting in less profit. Hence, it is an important KPI for the TOC. PTT can be calculated by summing over each timetable the number of passengers travelling between each pair of stations, at the time given in the timetable, multiplied by the time it takes them to travel that distance. The number of passengers travelling between the stations is given by a set of origin-destination matrices (ODM). One ODM is valid for exactly one timetable, but multiple ODMs from similar journeys can be aggregated for this article where only the busiest cycle is optimised. An ODM is an $N(N-1) / 2 \times 3$ matrix, where $N$ is the number of stations in the timetable. The columns represent the origin station, destination station, and the number of passengers travelling from the origin to the destination. Each row represents each opportunity a passenger has to travel in the timetable. Therefore, the direction of travel in the ODM must follow the timetable. The calculation of PTT is

Table 2
Notation used in the mathematical formulation.

| Sets |  |
| :---: | :---: |
| $\mathcal{J}$ | Set of all trains. |
| $\mathcal{T}_{i k}$ | Set of trains that go in the same direction as train $i$ on section $k$. Train $i$ is not included. |
| $S_{i k l}$ | Set of triples $(j, m, n)$ where train $j$ goes in the same direction as $i$ at the same station identified by sections $m$ and $n$, and $k$ and $l$ respectively. Train $i$ is not included. |
| $\mathcal{F}$ | Set of triples $(\alpha, i, k)$ that indicate if a specific time in the timetable is fixed for train $i$ on section $k . \alpha$ indicate if it is a arrival/departure time. |
| $\mathcal{K}_{i}$ | Set of sections visited by train $i$. |
| $\mathcal{R}_{i}$ | Set of tuples of chronologically ordered sections, $(k, l)$, both visited by the train $i$ between which train $i$ performs no stops. |
| $\mathcal{A}_{\text {i }}$ | Set of tuples of chronologically ordered adjacent sections identifying a station, $(k, l)$, for train $i$. |
| Indices |  |
| $i, j$ | A train. |
| $k, l, m, n$ | A section. |
| Parameters |  |
| $r_{i k}^{\min }\left(r_{i k}^{\text {max }}\right)$ | Minimum (maximum) time for train $i$ to run on section $k$. |
| $w_{i k l}^{\min }\left(w_{i k l}^{\max }\right)$ | Minimum (maximum) time for train $i$ to dwell at station between sections $k$ and $l$. |
| $h_{k}^{\text {min }}$ | Minimum headway time on section $k$. |
| $T_{i k}^{a}\left(T_{i k}^{d}\right)$ | Original arrival (departure) time of train $i$ on section $k$. |
| $p_{i k l}$ | Number of passengers travelling with train $i$ between departure station on section $k$ to arrival station on section $l$. |
| $c_{k}$ | Capacity of section $k$ in one direction. |
| $c_{i k l}$ | Capacity of station between sections $k$ and $l$ on the same platform group and direction as train $i$. |
| M | An arbitrary very large positive number. |
| $\epsilon$ | An arbitrary very small positive real number. |
| Decision variables |  |
| $t_{i k}^{a}\left(t_{i k}^{d}\right)$ | Arrival (departure) time of train $i$ on section $k$. |
| $q_{i k j}^{a}\left(q_{i k j}^{d}\right)$ | 1 if train $i$ arrives (departs) before train $j$ arrives (departs) on section $k$ regardless of $h_{k}^{\text {min }}$. |
| $x_{i k l j}^{s}$ | 1 if train $i$ departs less than $h_{l}^{\min }$ before train $j$ arrives at station between sections $k$ and $l$. |
| $x_{i k j}^{a}\left(x_{i k j}^{d}\right)$ | 1 if train $i$ arrives (departs) less than $h_{k}^{\text {min }}$ before train $j$ arrives (departs) on section $k$. |
| $y_{i k l j}^{s}$ | 1 if trains $i$ and $j$ are dwelling at the same platform group at the station between sections $k$ and $l$ within $h_{l}^{\min }$ time of each other. |
| $y_{i k j}^{a}\left(y_{i k j}^{d}\right)$ | 1 if train $i$ arrives (departs) within $h_{k}^{\min }$ time of train $j$ arrives (departs) on section $k$. |
| $z_{i k l}^{s}$ | 1 if train $i$ dwell at the station between sections $k$ and $l$ at the same time as more trains than the platform group capacity. |
| $z_{i k}^{a}\left(z_{i k}^{d}\right)$ | 1 if the arrival (departure) of train $i$ overlaps with more trains than the section capacity. |

defined in Eq. (2) and consists of the running time and dwell time of the train. A more sophisticated PTT objective would consider entire passenger journeys, passenger train transfer times, and the change in passenger journeys and their choice of train services when the timetables change. However, that is not included in this formulation. The used notation for the mathematical formulation is presented in Table 2.

$$
\begin{equation*}
P(t)=\sum_{i \in \mathcal{T}}\left(\sum_{(k, l) \in \mathcal{K}_{i}} p_{i k l}\left(t_{i l}^{a}-t_{i k}^{d}\right)+\sum_{(k, l) \in \mathcal{A}_{i}} p_{i k l}\left(t_{i l}^{d}-t_{i k}^{a}\right)\right) \tag{2}
\end{equation*}
$$

This article proposes a new data-driven approach to estimating energy consumption for timetable optimisation of an entire network based on the work of Madsen et al. (2019). The used data consists of exact GPS positions, arrival/departure times, and speeds. This data makes it easy to infer the actual speed profile of a train running a section from which the estimated energy consumption can be derived. As it is nearly impossible for a human driver always accurately to adhere to a timetable, the speed profile and the amount of time spent running the section may differ from the advised timetable and planned section run time. By collecting all the train runs that have run the same sections with the same timetable, a relation can be learnt between the actual running time between stop stations and the consumed energy. More details on this method are described in Section 4. This relation is modelled as a non-increasing second-degree polynomial function and is referred to as the energy function. The sections between each neighbouring stop station have a different energy function, and the sum of energy functions is the energy consumption objective. This is shown in Eq. (3). Note, an energy function may span multiple sections, as the train may not stop at all stations it visits on its journey.

$$
\begin{equation*}
E(t)=\sum_{i \in \mathcal{T}} \sum_{(k, l) \in \mathcal{R}_{i}} e_{i k l}\left(t_{i l}^{a}-t_{i k}^{d}\right), \tag{3}
\end{equation*}
$$

where $e_{i k l}(\cdot)$ is the energy function for train $i$ departing on section $k$ and arriving on section $l$. The energy functions reveal which sections require more energy to run, i.e. which sections have a steeper polynomial. It is possible to save energy by moving slack time from less to more energy-consuming sections. Ideally, all sections get the maximal amount of time to run to be energy efficient. However, the timetables have several constraints to conform to. In addition to that, we have a competing objective, PTT, where the ideal run time is the minimal run time. Many other objectives exist to measure the attractiveness of the service, including punctuality and timetable robustness. However, working with more than two objectives is often impractical.

### 3.5. Mathematical formulation

Until now, the problem and its domain have been presented. This section will formalise these into a mathematical formulation for the multi-objective network-optimised energy-efficient train timetabling (MONO-EETT) problem, which is the version of TTP this article will consider. The mathematical model is based on continuous arrival and departure times and indicator variables introduced later. The model has an explicit representation of sections given by the indices $k$ and $l$ but has an implicit representation of stations. Thus, stations are represented by two sections in the network between which the station lies. There may be more than one pair of sections that identifies a station, but a pair of sections can only identify one station. This simple representation is possible because trains are assumed not to take capacity at start and end stations and that trains can only visit the same section once. The model does not guarantee the ordering of the trains which may cause overtakings to be added or removed during the optimisation. However, the route of each train in the network is fixed. The MONO-EETT problem is given by Eqs. (4)-(26), though, the constraints are not exhaustive compared to a real scenario and may not be directly applicable as the solution needs adjustment from the timetable planner. For example, adjustments are needed to ensure transfer possibilities with other trains, to arrival and departure times to ease communication to the passengers and to ensure trains do not collide on sections with bidirectional traffic.

In the below formulation, the authors chose to use the $\wedge$ symbol as a shorthand syntax for the logical conjunction of two binary decision variables. Eq. (4) gives the set objectives of the MONO-EETT problem. Constraint (26) forces $q$, $x, y$, and $z$ to be binary. Constraint (5) ensures that fixed arrival and departure times in the timetable will not change from the original time in the timetable. The minimum and maximum run time are enforced in Constraints (6) and (7). The absolute minimum run times are given by the physical limitations of the train and the infrastructure and varies with the specific locomotive's ability to brake and accelerate. In this article, minimum and maximum run times are set to be the fastest and slowest section runs respectively in the available historical data. This way we ensure to only optimise within the boundary of the data used for learning the energy functions. The minimum time needed to dwell at a station is given by Constraint (8) and is based on the TOC's experience with how much time is needed to load and unload passengers. Rush hours and bigger stations usually require higher minimum dwell times. For stations that the train will only pass through, the minimum dwell time will be 0 . Constraint (9) gives the maximum dwell time and will by default be set to the maximal integer value, i.e. positive infinity. One exception is trains passing through where the maximum dwell time will be 0 .

$$
\begin{equation*}
\text { minimise } \quad[E(t), P(t)] \tag{4}
\end{equation*}
$$

subject to

$$
\begin{align*}
& t_{i k}^{\alpha}=T_{i k}^{\alpha}, \quad \forall(\alpha, i, k) \in \mathcal{F}  \tag{5}\\
& t_{i k}^{a}-t_{i k}^{d} \geq r_{i k}^{\min }, \quad \forall i \in \mathcal{T}, k \in \mathcal{K}_{i}  \tag{6}\\
& t_{i k}^{a}-t_{i k}^{d} \leq r_{i k}^{\max }, \quad \forall i \in \mathcal{T}, k \in \mathcal{K}_{i}  \tag{7}\\
& t_{i l}^{d}-t_{i k}^{a} \geq w_{i k l}^{\min }, \quad \forall i \in \mathcal{T},(k, l) \in \mathcal{A}_{i}  \tag{8}\\
& t_{i l}^{d}-t_{i k}^{a} \leq w_{i k l}^{\max }, \quad \forall i \in \mathcal{T},(k, l) \in \mathcal{A}_{i}  \tag{9}\\
& t_{i l}^{d}+h_{l}^{m i n}-t_{j m}^{a} \leq M x_{i k l j}^{s}, \quad \forall i \in \mathcal{T},(j, m, n) \in S_{i k l},(k, l) \in \mathcal{A}_{i}  \tag{10}\\
& t_{j m}^{a}-\left(t_{i l}^{d}+h_{l}^{\min }+\epsilon\right) \leq M\left(1-x_{i k l j}^{s}\right), \quad \forall i \in \mathcal{T},(j, m, n) \in S_{i k l},(k, l) \in \mathcal{A}_{i}  \tag{11}\\
& y_{i k l j}^{s}=x_{i k l j}^{s} \wedge x_{j m n i}^{s}, \quad \forall i \in \mathcal{T},(j, m, n) \in S_{i k l},(k, l) \in \mathcal{A}_{i}  \tag{12}\\
& 1+\sum_{(j, m, n) \in S_{i k l}} y_{i k l j}^{s}-c_{i k l} \leq M z_{i k l}^{s}, \quad \forall i \in \mathcal{T},(k, l) \in \mathcal{A}_{i}  \tag{13}\\
& c_{i k l}-\left(1+\sum_{(j, m, n) \in S_{i k l}} y_{i k l j}^{s}+\epsilon\right) \leq M\left(1-z_{i k l}^{s}\right), \quad \forall i \in \mathcal{T},(k, l) \in \mathcal{A}_{i}  \tag{14}\\
& 2 \leq z_{i k l}^{s}+z_{j m n}^{s}+y_{i k l j}^{s}, \quad \forall i \in \mathcal{T},(j, m, n) \in S_{i k l},(k, l) \in \mathcal{A}_{i}  \tag{15}\\
& t_{j k}^{\alpha}+h_{k}^{\min }-t_{i k}^{\alpha} \leq M x_{i k j}^{\alpha}, \quad \forall i \in \mathcal{T}, j \in \mathcal{T}_{i k}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\}  \tag{16}\\
& t_{i k}^{\alpha}-\left(t_{j k}^{\alpha}+h_{k}^{\min }+\epsilon\right) \leq M\left(1-x_{i k j}^{\alpha}\right), \quad \forall i \in \mathcal{T}, j \in \mathcal{T}_{i k}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\}  \tag{17}\\
& y_{i k j}^{\alpha}=x_{i k j}^{\alpha} \wedge x_{j k i}^{\alpha}, \quad \forall i \in \mathcal{T}, j \in \mathcal{T}_{i k}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\}  \tag{18}\\
& 1+\sum_{j \in \mathcal{T}_{i k}} y_{i k j}^{\alpha}-c_{k} \leq M z_{i k}^{\alpha}, \quad \forall i \in \mathcal{T}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\}  \tag{19}\\
& c_{k}-\left(1+\sum_{j \in \mathcal{T}_{i k}} y_{i k j}^{\alpha}+\epsilon\right) \leq M\left(1-z_{i k}^{\alpha}\right), \quad \forall i \in \mathcal{T}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\}  \tag{20}\\
& 2 \leq z_{i k}^{\alpha}+z_{j k}^{\alpha}+y_{i k j}^{\alpha}, \quad \forall i \in \mathcal{T}, j \in \mathcal{T}_{i k}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\} \tag{21}
\end{align*}
$$

$$
\begin{align*}
t_{j k}^{\alpha}-t_{i k}^{\alpha} \leq M q_{i k j}^{\alpha}, & \forall i \in \mathcal{T}, j \in \mathcal{T}_{i k}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\}  \tag{22}\\
t_{i k}^{\alpha}-\left(t_{j k}^{\alpha}+\epsilon\right) \leq M\left(1-q_{i k j}^{\alpha}\right), & \forall i \in \mathcal{T}, j \in \mathcal{T}_{i k}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\}  \tag{23}\\
\sum_{j \in \mathcal{T}_{i k}} q_{i k j}^{a} \leq \sum_{j \in \mathcal{T}_{i k}} q_{i k j}^{d}+c_{k}-1, & \forall i \in \mathcal{T}, k \in \mathcal{K}_{i}  \tag{24}\\
\sum_{j \in \mathcal{T}_{i k}} q_{i k j}^{d} \leq \sum_{j \in \mathcal{T}_{i k}} q_{i k j}^{a}+c_{k}-1, & \forall i \in \mathcal{T}, k \in \mathcal{K}_{i}  \tag{25}\\
q_{i k j}^{\alpha}, x_{i k l j}^{s}, x_{i k j}^{\alpha}, y_{i k l j}^{s}, y_{i k j}^{\alpha}, z_{i k l}^{s}, z_{i k}^{\alpha} \in\{0,1\}, & \forall i \in \mathcal{T}, j \in \mathcal{T}_{i k}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\} \tag{26}
\end{align*}
$$

Constraints (10)-(15) are concerned with capacity and headway constraints at stations. Similarly, Constraints (16)-(21) are concerned with sections. The cornerstone of checking capacity constraints at stations and sections is interval overlap checks. This is easiest done by evaluating $s_{1} \leq e_{2} \wedge s_{2} \leq e_{1}$, where $s$ is the start time and $e$ is the end time for the intervals 1 and 2 . If the expression evaluates to true, the intervals overlap. This can be broken down into two less-than-or-equals expressions. At a station, two trains' timetables overlap, if one train arrive at the station before the other has departed plus the minimum headway time. This overlap check is represented by the variable $y_{i k l j}^{s}$ which is 1 if timetables of trains $i$ and $j$ overlap. The value of $y_{i k l j}^{s}$ is enforced by Constraint (12), which is the conjunction of the two less-than-or-equal expression ( $x_{i k l j}^{s}$ and $x_{j m n i}^{s}$ ) both being assigned by a value in the Constraints (10) and (11). Constraint (10) ensures that $x_{i k l j}^{s}$ is 1 if train $i$ arrives at the station (which for $i$ lies between sections $k$ and $l$ ) before $j$ leaves the station (which for $j$ lies between sections $m$ and $n$ ). Section indices $m$ and $n$ are introduced as train $j$ can enter and leave the station on different sections than train $i$ and still have overlapping dwell time. One example is when a platform on a station or a junction leads to two different stations. This is also shown in Fig. 3. Train $j$ is taken from the set $S_{i k l}$ implying that $j$ must use the same platform group in the same direction as $i$ at the station between sections $k$ and $l$. Constraint (11) ensures that $x_{i k l j}^{s}$ is 0 otherwise. If both $x_{i k l j}^{s}$ and $x_{j m n i}^{s}$ are 1 , an overlap has been detected and $y_{i k l j}^{s}$ is 1 . Similar checks are performed when a train enter and leave a section when checking section capacity in Constraints (16)-(18). The only difference is that running a section is not a blocking operation. Thus, the blocking train does not need to leave the section for another train to enter. Instead, we check if their arrival and departure times are at least $h_{k}^{\min }$ apart, respectively.

Constraint (13) ensures that $z_{i k l}^{s}$ is 1 if the dwelling of train $i$ at the station overlaps with more trains than the platform group capacity $\left(c_{i k l}\right)$ allows by summing the $y_{i k l j}^{s}$ variables. Note, that the capacity is given for one direction only because it is identical for both directions as mentioned earlier. $c_{i k l}=1$ means that there is a track in both directions, i.e. 2 in total. Constraint (14) ensures that $z_{i k l}^{s}$ is 0 otherwise. This count alone is not enough to determine a capacity violation, so, we define a capacity violation between two trains if 3 conditions are met: (1) train $i$ overlaps with more trains than the capacity ( $z_{i k l}^{s}$ ), (2) train $j$ overlaps with more trains than the capacity $\left(z_{j m n}^{s}\right)$, and (3) trains $i$ and $j$ overlap ( $y_{i k l j}^{s}$ ). This is enforced by Constraint (15). Analogously, the same constraints exist for when a train enters and leaves a section - no more trains can drive onto a section or leave a section than the section capacity allows, unless the minimum headway time is respected.

Constraints (22)-(25) avoid illegal overtakings. The variable $q_{i k j}^{\alpha}$ indicates if train $i$ precedes train $j$ on arrival/departure at section $k$, this is ensured by Constraints (22) and (23). Next, Constraint (24) then ensure that a train can only take over the number of other trains which the capacity allows. For example, if $c_{k}=1$, then the train cannot overtake at all. If $c_{k}=2$, then the train can overtake one train on that section. This is to ensure that a train does not try to overtake two other trains that runs exactly parallel when $c_{k}=2$. Similarly, Constraint (25) ensures that a train cannot be overtaken by too many trains at a time by the same principles.

Other research (Wang et al., 2019) uses big M for capacity and headway constraints similarly to Constraints (16) and (17). Differently to this article, they do not aggregate the platform capacity (e.g. for a platform group) and checks for each platform if the headway time is respected.

## 4. A data-driven matheuristic for MONO-EETT

This section will introduce our proposed algorithm for solving the MONO-EETT problem defined in Section 3.5. The algorithm is a matheuristic using a mathematical program to generate the initial population for a GA based on NSGA-II (Deb et al., 2002). Both use two objectives: PTT and energy consumption. The latter uses a novel approach to energy consumption estimation by learning the trade-off between energy consumption and running time from historical data as a preparation step to the matheuristic.

### 4.1. Learning the trade-off curve from historical data

An important part of this article is to model the relationship between actual energy consumption and running time for each section in the network from extensive historical data. Contrary to the well established approach in literature explained in Section 2 , the trade-off curve will instead be estimated based on actual data rather than simulated through EETC. One of the advantages of doing so is the accurate modelling of the actual driving behaviours on the network. In previous literature this could only be achieved, for instance, by adding fuzzy variables to the EETC simulation when estimating the trade-off curve. Since the outcome of the two approaches are the same (i.e. a trade-off curve), both approaches could in future literature be used interchangeably, depending on the desired outcome of the timetable optimisation problem at hand. A comparison between the two approaches will be given at the end of this section, in order to visually inspect the difference between the two obtained trade-off curves.

First, to model the trade-off curve, Madsen et al. (2019) proposes a five-step approach, where the first three steps are the following: (1) data reduction, (2) outlier detection, and (3) regression modelling. The same steps will be used in this article,


Fig. 5. Learning trade-off curve: data reduction, outlier detection, and regression modelling steps.


Fig. 6. Comparison between the trade-off curves obtained by fitting to actual data and simulating data respectively.
however, with some improvements. First, instead of modelling the energy consumption of each specific train configuration, the energy consumption is divided by the total mass of the train to get the energy consumption of moving a ton ( $\mathrm{Wh} / \mathrm{t}$ ). That increases the amount of data available for each section, contributing to a more accurate fit in regression modelling. Second, to handle fluctuating energy consumptions for different running times, Madsen et al. (2019) suggested rounding real-valued run times to nearest integer and average their energy consumption. However, our experiments have shown that a better fit in regression modelling was obtained using the median instead. The outlier detection step has not changed, since the density-based clustering algorithm DBSCAN (Ester et al., 1996) performs satisfactory with the parameters ( $\varepsilon=0.08$, MinPts $=10$ ) obtained in Madsen et al. (2019). Last, an energy function was fitted for each planned running time in the regression modelling step. Thus, the same section could have multiple energy functions based on different timetables. Naturally, the only difference between those energy functions is a displacement in time due to the different planned target arrival times. Since the slopes should be similar, the distinction in planned run time is unnecessary added complexity. This article proposes to calculate the average energy function, to generalise the multiple energy functions for each section into one. The final average energy function obtained is equivalent to a trade-off curve used in the literature.

Fig. 5 shows the results in each step of the improved approach, which can be compared to the previous approach in Madsen et al. (2019). Fig. 5(a) shows the raw historical data of every planned trip conducted between two stops in the network. Fig. 5(b) shows the output of the three first steps for one out of six planned run times. Last, Fig. 5(c) shows the fitted energy functions for each planned run time on a section, and the final average energy function, $e_{i k l}(\cdot)$ (dotted black). Using the suggested approach, a satisfactory fit was obtained on all instances of the historical data. Specific to Fig. 5(b), a $R^{2}$ value of 0.93 was obtained.

As explained in the beginning of this section, a rather traditional approach to estimating the trade-off curves exists. Fig. 6 shows two trade-off curves. The green curve is the result of the data-driven approach proposed by Madsen et al. (2019), and is the same trade-off curve obtained in Fig. 5. The blue curve is the trade-off curve obtained by using the well-established approach in literature (Scheepmaker et al., 2017). The EETC simulations are obtained by using the proprietary solver by Cubris. The first obvious difference between the two curves is before the 230 s mark. This inaccuracy is due to missing or sparse data before 230 s and after 370 s , as seen on Fig. 5(b). These inaccuracies are handled by limiting the temporal range of the trade-off curve. More relevant to the
comparison is the noticeable difference in energy consumption overall. As expected, the actual energy consumption is much higher than the simulated energy consumption, due to the difference in driving behaviours and the human factor. The blue curve shows what ideally is to be achieved, whereas the green curve shows what is more likely to be achieved on the network. The simulated trade-off curve can be seen as an optimistic model, whereas the actual trade-off curve to a greater extent models the realised behaviour. Using the new proposed data-driven approach gives future research new possibilities, for instance, of benchmarking the potential energy savings when combining EETC with EETT.

### 4.2. The matheuristic

In the literature review it was evident that heuristic approaches to TTP has gained a lot of attention. Heuristic approaches, such as GAs, allow finding good approximate solutions by exploring large search spaces within reasonable time. Another significant benefit is the ability to optimise multiple objectives without needing to quantify their importance or weight before optimisation. In the MONO-EETT problem, $E(t)$ and $P(t)$ are incomparable and competing, i.e. reducing one will increase the other. Non-domination based algorithms, such as NSGA-II, handles this. For these reasons, this article proceeds with a GA as the foundation for the proposed matheuristic to solve the MONO-EETT problem.

```
Algorithm 1: Outline of the matheuristic based on NSGA-II.
    def optimise( \(T\) : Original timetables, \(P\) : Population size):
        \(\mathcal{P} \leftarrow\) MIP-initialiser \((T, P) \quad \triangleright\) Algorithm 2
        while \(\neg\) terminate_algorithm() do
            \(\mathcal{Q} \leftarrow\) empty set of offspring
            for \(i \leftarrow 1\) to \(P\) by 1 do
                \((r, p) \leftarrow\) binary_tournament_selection \((\mathcal{P})\)
                \(q \leftarrow\) hybrid_single_point_crossover \((r, p)\)
                \(q^{\prime} \leftarrow \operatorname{mutation}(q)\)
                \(w \leftarrow\) repair_heuristic \(\left(q^{\prime}\right) \quad \triangleright\) Algorithms 3-5
                if feasible (w) then
                \(\mathcal{Q} \leftarrow \mathcal{Q} \cup\{w\}\)
            end
            end
            \(\mathcal{R} \leftarrow \mathcal{P} \cup \mathcal{Q}\)
            assign_non_dominated_rank( \(\mathcal{R}\) )
            assign_crowding_distance \((\mathcal{R})\)
            \(\mathcal{P} \leftarrow\) survivor_selection \((\mathcal{R})\)
        end
        return \(\mathcal{P}\)
```

Algorithm 1 provides an overview of the proposed matheuristic. Based on the original timetables $T$, an initial population $\mathcal{P}$ of size $P$ is created on line 2 using the MIP-initialiser. Lines $3-18$ comprises the GA approach to optimisation of the population. The optimisation works by generating $P$ children on lines 5-13 which represents one generation of offspring. To decrease the running time per generation, the loop is parallelised to produce multiple children at a time. First, two parents are selected on line 6 (see Section 4.2.4). Then a child is created by combining the two parents through crossover on line 7 (see Section 4.2.5). To ensure diversity from the rest of the population, the child is mutated on line 8 (see Section 4.2.6). In this article, all individuals in the population must be feasible. Therefore, the child is repaired on line 9 to guarantee that all constraints are met (see Section 4.2.7). Only if the repair succeeded, the child is added to the set of offspring on line 11.

When the set of feasible offspring has been generated, they get added to the population on line 14 . Then, each individual is assigned a domination rank and a crowding distance on lines 15 and 16 (see Section 4.2.1). Last, these two metrics are used to select the fittest $P$ individuals on line 17 based on non-dominated sorting (see Eq. (27) of Section 4.2.1).

### 4.2.1. NSGA-II

The matheuristic presented in this article is based on NSGA-II and uses the same sorting that primarily sorts the population by the non-domination rank of an individual and secondarily sorts by crowding distance (in descending order). The non-domination rank is the number of other individuals a given individual is dominated by. An individual is dominated by another individual if the other individual has strictly better objective values for all objectives. The crowding distance is sum of the normalised distances to neighbouring individuals on the Pareto front. By these two metrics we can define an order of individuals to sort them by their fitness (Deb et al., 2002):

$$
\begin{equation*}
p<q: \operatorname{rank}(p)<\operatorname{rank}(q) \vee(\operatorname{rank}(p)=\operatorname{rank}(q) \wedge \operatorname{dist}(p)>\operatorname{dist}(q)) \tag{27}
\end{equation*}
$$

where $\operatorname{rank}(p)$ is the non-domination rank of $p$ and $\operatorname{dist}(p)$ is the crowding distance of $p$. Non-domination and crowding distance ensures both quality and diversity of the population. The reader is referred to Deb et al. (2002) for a in-depth explanation of the metrics and sorting.

### 4.2.2. Genotypic representation

In the genotypic representation implemented in this article, one chromosome represents a potential solution to the set of timetables in the network. Each chromosome describes an individual in the population. In the chromosome, the genes are organised in two dimensions, where a real-valued representation of genes has been chosen. The first dimension contains the timetables, where each index $i$ is a timetable for a train line. The second dimension contains the timetable sections, where each index $k$ is a section in the timetable $i$. One gene, therefore, consists of the arrival and departure times for the timetable $i$ on section $k$. An example of the chromosome is shown in Fig. 7. This structure allows for easier access for the GA operators when handling multiple timetables. Compared to single-vector representations (such as Xu et al., 2016; Wang et al., 2019) typically used for single timetable optimisation, the representation is less memory efficient as the lists of genes are not equal in length. Both representations give a 1 -to- 1 correspondence between genes and the decision variables used in the MONO-EETT formulation. The $i$ th timetable from the chromosome will be used where applicable instead of the whole chromosome to simplify visual examples later in the article because operations are performed equally on each timetable. The $i$ th timetable will be represented as the usual horizontal value vector. This is shown in Fig. 7.

### 4.2.3. Initial population

GAs are sensitive to the initial population in terms of diversity of solutions, convergence, and quality of solutions. Our approach to generating the initial population is inspired by the MIP-recombination proposed by Borisovsky et al. (2009) for solving the supply management problem. Using this method requires the problem at hand to be solved by a MIP solver in a short amount of time.

We will use a similar approach to warm-start the GA with an initial population consisting of high quality and diverse solutions. We call this a MIP-initialiser. In this article, the entire initial population besides the original timetable is generated by the MIP-initialiser. When the term "locking" is used, we refer to Constraint (5). Instead of operating on two parents, this MIP-initialiser will operate using one parent being the original timetable. The advantage of using the MIP-initialiser is that no constraints are being violated. Using the MIP-initialiser is only possible if the problem takes a relatively short amount of time to solve by the MIP solver. It is shown in Section 5.3 how the initiation time grows with the number of variables to optimise. If this time grows too big for the problem at hand, heuristic methods are a viable option. Here, it is important to judge the trade-off between computation time and solution quality when comparing alternative initialisation methods. For instance, benchmarks with the two metrics, Hypervolume (Yu et al., 2018) and $\Delta$ (Deb et al., 2002), introduced in Section 5 could be used for this. Given the original timetables $T$ and the population size $P$, the MIP-initialiser goes through the steps detailed in Algorithm 2.

```
Algorithm 2: Outline of MIP-initialiser.
    def MIP-initialiser( \(T\) : Original timetable, \(P\) : Population size):
        \(\mathcal{P} \leftarrow\{T\}\)
        \(\mathcal{P} \leftarrow \mathcal{P} \cup\{T\) optimised with respect to \(E(t)\}\)
        \(\mathcal{P} \leftarrow \mathcal{P} \cup\{T\) optimised with respect to \(P(t)\}\)
        for \(i \leftarrow 1\) to \(P-2\) by 1 do
            \(O(\cdot) \leftarrow E(\cdot)\) or \(P(\cdot)\) by uniform probability
            if \(O(\cdot)=E(\cdot)\) then
                \(p \leftarrow i /(P-2) \quad \triangleright\) referred to as \(p_{E}\)
            else if \(O(\cdot)=P(\cdot)\) then
                \(p \leftarrow 0.3 i /(P-2) \quad \triangleright\) referred to as \(p_{P}\)
            \(\mathcal{F} \leftarrow \emptyset\)
            for \(\alpha \in\{a, d\}, i \in \mathcal{T}, k \in \mathcal{K}_{i}\) do
                if with probability \(p\) then
                    \(\mathcal{F} \leftarrow \mathcal{F} \cup(\alpha, i, k)\)
            end
            \(\mathcal{P} \leftarrow \mathcal{P} \cup\{T\) optimised with respect to \(O(t)\) using fixation constraints based on \(\mathcal{F}\}\)
        end
        return \(\mathcal{P}\)
```

The initial population will consist of the original timetable, a PTT-optimal timetable, and an energy-optimal timetable. This is shown on lines $2-4$. Furthermore, $P-3$ individuals are generated on lines $5-16$ by randomly locking times in the timetables. When the optimal single-objective individuals are added to the initial population, the search is more easily guided to find solutions dominating more of the solution space. All generated individuals are added to the set $\mathcal{P}$ and returned.

The two objectives use different probabilities, $p_{E}$ and $p_{P}$, for locking times in the timetables based on our experimental results (see Section 5). This is reflected on lines 8 and 10. A straightforward approach would be to assign a flat probability to $p_{E}$ and $p_{P}$ respectively, i.e. it is the same for all individuals. A higher probability for locking a time shifts the generated individuals closer to the original timetable. A lower probability shifts the generated individuals closer to the optimal solutions for each objective. However, this does not contribute to diverse trade-offs between the objectives. A flat probability only produce trade-offs in the same neighbourhood in the solution space. This is shown in Section 5. Instead, we made the probability depend on the index, $i$, of


Fig. 7. A chromosome in the GA.
the individual in the initial population. This way, individuals are created with different trade-offs between $E(t)$ and $P(t)$ extending from the optimal solutions to the original solution.

On lines $12-14$, the set of fixed arrival and departure times in the timetable, $\mathcal{F}$, is constructed. $\mathcal{F}$ contains triples of $\alpha$, the train $i$ and the section $k . \alpha$ indicates with $a$ if it is an arrival time and $d$ if it is a departure time. This is also summarised in Table 2.

The MIP model optimised on lines 3, 4, and 16 is based on the constraints of MONO-EETT given in Section 3 with three minor modifications. First, only one objective, $O(\cdot)$, will be optimised at a time. Second, Constraints (28) and (29) are added to the formulation.

$$
\begin{array}{ll}
t_{i k}^{\alpha}-T_{i k}^{\alpha} \leq d^{\max }, & \forall i \in \mathcal{T}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\} \\
T_{i k}^{\alpha}-t_{i k}^{\alpha} \leq d^{\max }, & \forall i \in \mathcal{T}, k \in \mathcal{K}_{i}, \alpha \in\{a, d\} \tag{29}
\end{array}
$$

These constraints ensure that each arrival and departure time in the timetable cannot change more than $d^{\max }$ from the original timetable. Otherwise, the MIP solver may move a timetable arbitrarily much into the future to avoid conflicts. We set $d^{\max }=300 \mathrm{~s}$ to keep changes small.

The third change is the linearisation of $E(t) . E(t)$ is a convex quadratic objective function and takes slightly more computational effort to optimise than a linear objective. We experienced that the linearised objective resulted in $10 \%-15 \%$ faster computation times than its quadratic counterpart. Therefore, this objective has been linearised for the MIP-initialiser by five linear tangents. A new decision variable is introduced, $e_{i k l}^{\prime}$, and Constraint (30) is added:

$$
\begin{equation*}
e_{i k l}^{\prime} \geq \lambda_{i k l 1}^{f}\left(t_{i l}^{a}-t_{i k}^{d}\right)+\lambda_{i k l 0}^{f}, \quad \forall i \in \mathcal{T},(k, l) \in \mathcal{A}_{i}, f \in\{1, \ldots, 5\}, \tag{30}
\end{equation*}
$$

where $\lambda_{i k l 1}^{f}$ is the linear coefficient and $\lambda_{i k l 0}^{f}$ is the constant of the $f$ th linearisation function. $e_{i k l}^{\prime}$ will now represent the energy consumption of train $i$ departing from section $k$ to section $l$. Finally, the linearised objective becomes:

$$
\begin{equation*}
E^{\prime}(t)=\sum_{i \in \mathcal{T}} \sum_{(k, l) \in \mathcal{A}_{i}} e_{i k l}^{\prime}, \tag{31}
\end{equation*}
$$

This showed a minor improvement in the MIP-initialiser's computational performance. It also allows for optimisation beyond the global minimum of the energy function, assuming that the energy will neither increase nor decrease. This is relevant when the maximum section run time lies beyond the global minimum of the second-degree polynomial.

### 4.2.4. Parent selection

Selecting parents to generate a child through crossover comes with a trade-off between quality and diversity. This implementation uses the binary tournament selection from NSGA-II, where two individuals are picked randomly from the population. Two parents are needed to create an offspring, thus two tournaments are needed. For each tournament, only the fittest parent is selected according to the sorting introduced in Eq. (27). After the tournaments, the two fittest parents will create an offspring through crossover and mutation. Parent selection is repeated for each crossover.

### 4.2.5. Crossover

Two crossovers have been considered in this article: single-point crossover (SPC) (Mitchell, 1998) and average crossover (AC) (Abd Rahman and Ramli, 2013). Both crossovers have their strengths and weaknesses and will be studied further in Section 5. SPC works by choosing a random station $k$ in a timetable $i$ to be the crossover point. Then the child of the crossover gets sections $1, \ldots, k-1$ from the first parent, and the sections $k, \ldots,\left|T_{i}\right|$ from the second parent. $\left|T_{i}\right|$ is the number of sections in timetable $i$. This is demonstrated in Fig. 8.

In the work of Abd Rahman and Ramli (2013), the AC generates two children and works by using a similar crossover point to define how many genes are being averaged. The first child gets its first half (until the crossover point) averaged and the second


Fig. 8. Example of single-point crossover.


Fig. 9. Example of average crossover.
half is unchanged. The second child gets its second half averaged and the first half is unchanged. Inspired by this, the authors of this article chose to average the entire chromosome to produce one child. This is shown in Fig. 9. For AC, the crossover operation affects all timetables, while SPC only changes one timetable. This configuration of the crossover operators proved the best results in our experimental tuning of the operators.

Neither SPC nor AC guarantees a conflict-free crossover. Since the crossover impacts the arrival and departure times, constraints could have been violated. For instance, the random station crossover point in SPC could have gotten an arrival time which is after its departure time. Station capacities could have been violated for both SPC and AC if too much dwell time was given, and the minimum and maximum run time constraints could have been violated. Therefore, repair heuristic described in Section 4.2 .7 are employed to repair capacity conflicts in the generated children after both crossover and mutation to ensure feasibility throughout the algorithm's running time. Unfortunately, the literature review did not uncover the details of repair strategies in the relevant papers. This repair heuristic is not guaranteed to be successful, thus, the population is not guaranteed to produce $P$ new offspring each generation. Though, this may decrease diversity of the population over time. This is in contrast to thorough repairs guaranteeing $P$ offspring which comes at the cost of running time.

This article proceeds with a hybrid crossover of AC and SPC, where each crossover has a $50 \%$ chance of being selected. Experimental evidence is shown in Section 5. Neither SPC nor AC alone performs as good as the hybrid.

### 4.2.6. Mutation

The mutation operator is quite simple but important for the diversity of the Pareto front and to help escape local minima. Each section in all timetables has a probability $p_{\text {mut }}$ to get its arrival and departure times changed by some random value. Therefore, multiple timetables can be mutated at a time, and a timetable can have multiple mutated sections. The random value is sampled from a 0 -mean Gaussian distribution with a standard deviation $\sigma_{m u t}$ : once for the arrival time and once for the departure time. This causes smaller mutations to be common and bigger mutations to be unlikely but not impossible. Big mutations are generally avoided for MONO-EETT because they are more likely to turn out infeasible. After the mutation, the constraints are repaired. If the repair fails, the mutated child is discarded, and the unmutated child proceeds. At this point, the unmutated child is not guaranteed to conform to the constraints. Therefore, the child will be repaired by the repair heuristic presented in Section 4.2.7. If the repair fails, the child is discarded and the next parent selection is started. For this implementation $p_{m u t}=0.02$ and $\sigma_{m u t}=10$ were chosen.

### 4.2.7. Repair heuristic and capacity conflict resolution

Considering the complexity of applying multiple crossovers and mutations on timetables in a heavily constrained network, a repair heuristic is required to guarantee feasible timetables. The design goal of the crossover and mutation operators described in Sections 4.2.5 and 4.2.6 has been to make fast, small, and local mutations to the timetables, in order to minimise the number of conflicts to be repaired by the heuristic. The repair heuristic does not guarantee any ordering of the timetables and may insert or remove overtakings if it is needed to make the timetables feasible. If the timetables have been destroyed too much, the repair can end up taking too much computation time, which in the end can cause less diverse solutions. This is opposed to using otherwise popular large neighbourhood search methods, which instead would require a non-greedy repair heuristic to ensure the survival of the diverse individuals.

To ensure fast computation time, the repair heuristic implements a greedy first-come-first-serve (FCFS) strategy with a fail-fast approach. Since individual timetable constraints are repaired after both the crossover and mutation operators, the heuristic repairs the network constraints consisting of overtaking constraints and station and section capacity constraints. The heuristic will try in a


(c) Repair strategy of section capacity violation.

Fig. 10. Repair heuristic strategies. The solid lines represent a solution with a violation, and the dashed lines represent the repaired solution.
given amount of iterations to repair the network. If unsuccessful, the child will be discarded. In each iteration, the heuristic applies a repair strategy for the overtaking constraint and the station and section capacity constraints. Each repair strategy will be described next.

The overtaking constraint repair heuristic outlined in Algorithm 3 works as follows. Note, the heuristic is only applied to the arriving trains at a station to avoid repairing the overtaking constraint violation multiple times to the same section. First, the trains are ordered by their departure time and arrival time on lines 2 and 3 . Then, the departure index $j$ and arrival index $k$ are retrieved for a train in order of their departure time on lines 5 and 6 . If the absolute difference between a departing train's departure and

```
Algorithm 3: Outline of fail-fast overtaking repair heuristic.
    def repair_section_overtaking ( \(k\) : section, \(\mathcal{T}^{=}\): trains going in same direction on section \(k\) ):
        \(\mathcal{T}^{d} \leftarrow \mathcal{T}=\) sorted by \(t_{i k}^{d}\)
        \(\mathcal{T}^{a} \leftarrow \mathcal{T}=\) sorted by \(t_{i k}^{a}\)
        for \(i \in \mathcal{T}^{d}\) do
            \(j \leftarrow\) index of \(i\) in \(\mathcal{T}^{d}\)
            \(k \leftarrow\) index of \(i\) in \(\mathcal{T}^{a}\)
            if \(|k-j| \geq c_{k}\) then
                \(t_{i k}^{\prime a} \leftarrow t_{j k}^{a}\)
                \(t_{j k}^{\prime a} \leftarrow t_{i k}^{a}\)
                if \(t_{i k}^{\prime a}-t_{i k}^{d} \geq r_{i k}^{m i n} \wedge t_{j k}^{\prime a}-t_{j k}^{d} \geq r_{j k}^{m i n} \wedge t_{i, k+1}^{d}-t_{i k}^{\prime a} \geq w_{i k, k+1}^{m i n}\) then
                    \(t_{i k}^{a} \leftarrow t_{i k}^{\prime a}\)
                    \(t_{j k}^{a} \leftarrow t_{j k}^{\prime a}\)
                else
                    return without changes (repair failed)
        end
        return with changes (repair succeeded)
```

arrival indices is greater than or equal to the section capacity $c_{k}$, then the train is involved in an overtaking constraint violation and thus must be repaired on lines $8-12$. The overtaking violation is repaired by switching arrival times with another train where the arrival index equals the train's departure index on lines 8 and 9 . It is only allowed to switch arrival times on lines 11 and 12 if it does not violate the minimum dwell time constraint $w_{i k, k+1}^{\min }$ and the minimum running time constraints $r_{i k}^{\min }$ and $r_{j k}^{\min }$. Otherwise, the repair has failed and the child is discarded based on the fail-fast strategy.

Fig. 10(a) illustrates this repair strategy, where the last departing train $v$ at station $a$ produces an overtaking constraint by arriving before train $i$ and $j$ at station $b$. Train $v$ has a departure index of 2 but an arrival index of 0 , thus, violating the section capacity $c_{k}$ of 2 . To repair this, $v$ first switches arrival time with $j$ and afterwards $j$ switches arrival time with $i$. The repaired individual then becomes $i^{\prime}, j^{\prime}$, and $v^{\prime}$.

Next, the station capacity repair heuristic will be considered which is outlined in Algorithm 4. First, the station capacity utilisation is calculated on line 3 as the number of overlapping dwelling trains with the current train, $i$. This is explained in Section 3.5 in Constraints (10)-(15). If the current station capacity utilisation is greater than or equal to the station capacity constraint $c_{i k l}$, then the station capacity constraint is violated. By the FCFS strategy, only the last violating trains will get allocated a new arrival time in order to repair the violation. Thus, line 5 filters and sorts the trains by their arrival time. Next, for each violating train in $\mathcal{T}^{\prime}$, the amount of drift required to repair the violation is calculated on line 7. This is calculated as the difference between the current overlapping train $j$ 's departure time and the current train $i$ 's arrival time plus the headway time $h_{l}^{\min }$. If the arrival time is not locked according to Constraint (5) and adding the drift to the current arrival time does not violate the maximum running time constraint $r_{i k}^{\max }$, then the arrival time is updated on line 9 . This way, the violating trains will arrive as soon as a platform is cleared and the headway time has been obeyed. Fig. 10(b) illustrates a simple example of how the station capacity violation gets resolved. The platform capacity at station $a$ is 2 but on station $b$ it is 1 , therefore train $i$ blocks the single platform until it departs plus a headway time. Train $j$ is scheduled to arrive at time $a_{j}$ in the blocking interval of train $i$ from $a_{i}$ to $d_{i}+h$. Train $j$ is therefore rescheduled to arrive when train $i$ has departed from the platform plus the headway time.

```
Algorithm 4: Outline of repairing station capacity using the FCFS strategy.
    def repair_station_capacity \(\left((k, l)\right.\) : station, \(\mathcal{T}^{=}\): trains going in same direction on station \((k, l)\) ):
        for \(i \in \mathcal{T}^{=}\)do
            overlaps \(\leftarrow\) count_overlaps_at_station( \(i, k, l\) )
            if overlaps \(\geq c_{i k l}\) then
                \(\mathcal{T}^{\prime}=\left\{j \in \mathcal{T}^{=} \mid t_{j k}^{a}>t_{i k}^{a}\right\}\) sorted by \(t_{j k}^{a}\) in descending order
            for \(j \in \mathcal{T}^{\prime}\) do
                \(\mathrm{drift}=t_{j l}^{d}+h_{l}^{\text {min }}-t_{i k}^{a}\)
                if \((a, i, k) \notin \mathcal{F} \wedge t_{i k}^{a}+\mathrm{drift}-t_{i k}^{d} \leq r_{i k}^{\max }\) then
                    \(t_{i k}^{a} \leftarrow t_{i k}^{a}+\mathrm{drift}\)
            end
        end
```

Last, the section capacity repair heuristic shown in Algorithm 5 will be described. First, the section capacity utilisation is calculated on line 3 as the number of overlapping departing or arriving trains using interval overlap checks for the headway time, as defined in Section 3.5 in Constraints (16)-(18). Similarly to the station capacity repair heuristic, the section capacity repair heuristic also implements the FCFS strategy. However, if the current section capacity utilisation is greater or equal to the capacity constraint $c_{k}$, then the last violating trains will get delayed at the station dwelling until the section is cleared. Thus, line 5 filters and sorts the trains by their departure time. Next, for each violating train in $\mathcal{T}^{\prime}$, the amount of drift required to repair the violation is calculated on line 7 . This is calculated as the difference between the current overlapping train $j$ 's departure time and the current train $i$ 's departure time plus the headway time $h_{k}^{\min }$. If the departure time is not locked according to Constraint (5) and adding the drift to the current departure time does not violate the minimum running time constraint $r_{i k}^{\min }$, then the departure time is updated on line 9. Fig. 10(c) illustrates a simple example of how the section capacity violation gets resolved. There is only a single track between station $a$ and $b$, therefore, train $i$ blocks train $j$ from departing until the headway time $h_{\text {min }}$ has passed. To repair this violation, train $j$ gets rescheduled to be dwelling at station $a$ until the headway time has passed.

If any constraints remain violated after all repair strategies have been attempted, the child is discarded.

## 5. Results - A case study

This section will argue for non-trivial parameter choices from an experimental standpoint and discuss how the network's complexity and degrees of freedom will affect the performance in terms of quality and computational time. Finally, the results in terms of the objectives will be presented and related to the business case. All results in this section are based on the case study presented in Section 5.1 to evaluate the combination of the data-driven approach with the matheuristic. This is challenging to evaluate properly on synthetic and random instances because EETC approaches to the estimation of the trade-off curve would then be more appropriate for such evaluation than the data-driven approach suggested in this article. For this reason, evaluation of the matheuristic on random instances has not been considered. The case study uses network infrastructure and energy consumption data
from regional trains running 9 train lines giving 18 timetables in the network. Due to their confidentiality, it is not possible to share the ODMs and related travel flow data. For simplicity, minimum headway and dwell time are equal for all timetables and sections, where $h_{k}^{\text {min }}=60 \mathrm{~s}$ and $w_{i k l}^{\min }=30 \mathrm{~s}$. Following parameters are set for the parameter tuning unless explicitly specified otherwise: population size is $P=50$, number of generations is 100 , mutation standard deviation is $\sigma_{\text {mut }}=10$ and mutation rate is $p_{\text {mut }}=0.02$. All tests have been run on a Windows 10 computer with AMD FX- $8320 @ 3.5 \mathrm{GHz}$ and 24 GB RAM. The MIP-initialiser uses the commercial MIP solver Gurobi 9.1 with standard parameters. The algorithm is implemented in the programming language C\# .NET Core 5.0.

For evaluating the quality of Pareto fronts, two metrics are used: Hypervolume (HV) (Yu et al., 2018) and $\Delta$ (Deb et al., 2002). In short, the HV metric indicates how much of the solution space is dominated by the Pareto front and if the algorithm has converged. The metric has the scale $[0,1]$, and a higher value is more desirable. The used reference point is created from the optimal solutions by using the maximum value of each objective. $\Delta$ is used to indicate diversity and convergence of the Pareto front. A value closer to 0 is more desirable. The reader is referred to the cited papers for a more in-depth explanation of these metrics.

### 5.1. Case study description

Until now, we have introduced the incorporation of historical data from rolling stock for energy consumption estimation. This subsection will introduce the case study that will be used to evaluate the proposed algorithm. The research is conducted in collaboration with Cubris that develops a DAS called GreenSpeed for European TOCs. A DAS is a unit installed in the driver's cabin giving real-time advice to the train driver for what speed to drive. In the case of GreenSpeed, given the timetable of the train, it will advise energy-efficient driving by calculating the energy-optimal speed profile using a proprietary algorithm (Haahr et al., 2017). In general, it is possible to save between $5 \%-20 \%$ energy by optimising speed profiles (Hansen and Pachl, 2014). However, a train can only drive as efficiently as the timetable allows it. Thus, this article explores how data from DAS' can be used to enhance the energy efficiency of the train operation by adjusting the timetable.

As a part of Cubris' work in optimising their customers' operation, this article had access to historical train runs from a large North European TOC which in this article has been anonymised. The TOC operates regional trains from one of the busiest railway stations in Europe, labelled station ML-1 in Fig. 11. Since 2015, the TOC's trains have been equipped with GreenSpeed and have recorded more than 140,000 train runs (i.e. executed timetables) which is approximately 1.06 million data points in total where each data point contains a section run time and actual energy consumption.

However, this article will only analyse a subset of this data related to 9 train lines on the Main Line (ML) and the South West (SW), South (S) and South East (SE) branches. The train lines are depicted in Fig. 11. Each line has 2 timetables, one in each direction, giving 18 timetables. A total of 107 stations and junctions exist in this subset of the network. Depending on the train line, it takes 1 to 3 h from the initial departure for a train to reach its terminal station. Each train line cycles every hour during the day and has at most 12 cycles each day. Timetable planners rarely consider larger instances than this at a time to avoid too big changes after the planning. The station ML-1 and station SW-27 are the two stations in the subset that are farthest apart with over 250 km in straight line distance.

### 5.2. Parameter tuning

Not all parameters are discussed in this section because they are believed to be trivial to tune. Out of the 18 timetables in the aforementioned case, 10 are locked and cannot be optimised. The unlocked timetables are those going to and from ML-26, S-25 and S-33, i.e. the Main Line and Branch S. The reader can consult Fig. 11. By locking other timetables this article achieved similar results. For graphs showing any of the two metrics, the results are an average of 10 runs. Before the 10 runs, an additional run is added to warm up the cache which is not included in the results.

```
Algorithm 5: Outline of repairing section capacity using the FCFS strategy.
    def repair_section_capacity ( \(k\) : section, \(\mathcal{T}^{=}\): trains going in same direction on section \(k\) ):
        for \(i \in \mathcal{T}^{=}\)do
            overlaps \(\leftarrow\) count_overlaps_on_section \((i, k)\)
            if overlaps \(\geq c_{k}\) then
            \(\mathcal{T}^{\prime}=\left\{j \in \mathcal{T}^{=} \mid t_{j k}^{d}>t_{i k}^{d}\right\}\) sorted by \(t_{j k}^{d}\) in descending order
            for \(j \in \mathcal{T}^{\prime}\) do
                \(\mathrm{drift}=t_{j k}^{d}+h_{k}^{\min }-t_{i k}^{d}\)
                if \((d, i, k) \notin \mathcal{F} \wedge t_{i k}^{d}+\mathrm{drift}-t_{i k}^{a} \geq r_{i k}^{m i n}\) then
                    \(t_{i k}^{d} \leftarrow t_{i k}^{d}+\mathrm{drift}\)
            end
        end
```



Fig. 11. Subset of the network managed by the TOC. Note that not all stations and junctions are included.


Fig. 12. Different configurations of the MIP-initialiser. Each plot has solutions from 10 runs.

### 5.2.1. Initial population

Adjusting the probability for locking a time in the timetable in the MIP-initialiser is not a trivial task. Three experiments were carried out, and the results are shown in Fig. 12. Each plot has 1000 points representing a solution in the solution space. The MIP-initialiser was for each configuration run 10 times, generating 100 solutions each time. Fig. 12(a) gives the best result visually and according to the metrics. In this case, the MIP-initialiser was set up as described in Section 4.2. In Fig. 12(b), both objectives has a probability of locking a time that increases from 0.0 to 1.0 . Finally, Fig. 12(c) shows the result when both objectives have a flat probability of 0.3 of locking a time in the timetable. In general, the more timetables that can be optimised, the greater the potential savings. As an initial population, having diverse solutions with different trade-offs between the objectives is ideal. This is visible from the three experiments, the metrics of which are shown under each plot in Fig. 12. With the changing probabilities of locking, solutions with varying levels of trade-offs are generated, thus being much more diverse. With the diversity also comes a larger HV of dominated solution space. The flat probability MIP-initialiser only generates solutions in the same trade-off neighbourhood because only so much time can be redistributed, thus, becoming less diverse. Secondly, it is evident that the two objectives can tolerate different probability ranges in order to be diverse. PTT is very dependent on how stations are connected. Thus, if a timetable's times at two busy stations become locked, the PTT between these stations will not change. In the ODMs used for this article, passengers travelling between busy stations are the main contributor to PTT, while minor stations only contribute very little. When increasing the locking probability, the biggest potential for improvement is removed by increasing the probability of the busy stations getting locked. That is the main reason that $P(t)$ operates better at a lower locking probability than $E(t)$, and these results may change significantly with other ODMs than those used here. For the case at hand, the MIP-initialiser produces satisfactory results.

### 5.2.2. Crossover

The crossover operator is one of the most crucial parts of the GA for quality and diversity. Thus, making an informed decision on which crossover to use is therefore important. Three heuristic crossovers are considered: single-point crossover (SPC), average crossover (AC), and a hybrid crossover (HC) where SPC and AC have 50\% chance of being applied. An experiment for each crossover is shown in Fig. 13. Especially from Fig. 13(a) it is evident that SPC and HC take more generations than AC to converge. At the 150th generation, SPC has not reached a Pareto front of higher quality than AC in terms of non-domination and diversity. AC converges very fast, but it does not converge to a particular good Pareto front compared to HC. HC combines the best of both the AC and SPC. It finds good solutions relatively fast due to AC but can push to even better Pareto fronts due to SPC. This is also visible from


Fig. 13. The performance of different crossovers. Metrics are averages of 10 runs.

Table 3
Test scenarios. "Opt. TT" = Optimised timetables. "No. Opt. TT" = Number of optimised timetables. The columns "Network" and "Opt. TT" refers to the network branches in Fig. 11.

| Scenario | Network | No. stations | Opt. TT | No. Opt. TT |
| :--- | :--- | :--- | :--- | :--- |
| 1 | ML | 25 | ML | $2 / 2$ |
| 2 | ML + S | 60 | ML | $2 / 8$ |
| 3 | ML + S | 60 | ML + S | $8 / 8$ |
| 4 | ML + S + SE | 80 | ML | $2 / 14$ |
| 5 | ML + S SE | 80 | ML + S | $8 / 14$ |
| 6 | ML + S + SE | 80 | ML + S SE | $14 / 14$ |
| 7 | Full | 107 | ML | $2 / 18$ |
| 8 | Full | 107 | ML + S | $8 / 18$ |
| 9 | Full | 107 | ML + S + SE | $14 / 18$ |
| 10 | Full | 107 | All | $18 / 18$ |

Fig. 13(c) that shows the Pareto fronts created by SPC, AC, and HC. HC finds a slightly better Pareto front than AC in the 150th generation. In Fig. 13(b), SPC fails to improve the diversity significantly over time. AC and HC manage to improve, however, HC is a bit worse due to using SPC. The bad diversity metrics also come to show in Fig. 13(c), where SPC has a big gap in its Pareto front, which is a big contribution to $\Delta$. HC has some minor holes in the Pareto front that is visible in its $\Delta$-value also. The ability of HC to keep improving the Pareto front beyond AC at the cost of some diversity is the main reason that HC was chosen for the solution approach.

### 5.3. Scalability and sensitivity analysis

When applying a solution method to a real-world problem, it is important to assess the performance with respect to changes in the input. This subsection will analyse the quality of the solutions as the network gets more complex (number of stations) and gets more degrees of freedom (number of unlocked times).

Table 3 shows 10 scenarios that represent problems ranging from an isolated single-line optimisation to optimising timetables in a complex network. The scenarios are based on the line diagram shown in Fig. 11 in Section 5.1. The simplest scenario is based on the Main Line (ML), and to add more degrees of freedom, other branches of the network (S, SE, and SW) are unlocked for optimisation. As mentioned earlier, the imagined use case is optimising a subset of the timetables in the network. The complexity of scenarios is increased by adding more of the network to the optimisation, thus, adding more timetables that impose constraints on the timetables being optimised. The results of the analysis are shown in Table 4. The results are generated by running the algorithm on the scenario 10 times terminating after 100 generations. All results from the same scenario are generated using the same initial population, except for the, average initiation time, which is the average of 10 runs of the MIP-initialiser independent of the other columns. Since the proposed matheuristic is a two-step hybrid between a MIP and a GA, the performance of solving the problem using the MIP-initialiser versus using the GA is shown by denoting the distinct average running time for each step respectively.

The results show that the algorithm can produce consistent results given the same initial population. The standard deviation of the metrics, $\sigma_{H V}$ and $\sigma_{\Delta}$, indicate a relatively small spread of the results of the 10 runs. Generally, when optimising a few timetables at a time (e.g. scenarios $1,2,4$, and 7 ), the dominated HV is usually smaller. Also, the average computation time of 100 generations is affected by the degrees of freedom. When more timetables are locked (i.e. fewer are optimised), the computation time goes up. This may be explained by the repair heuristic experiencing more failed attempts to repair a solution with fewer degrees of freedom. One example is the overtaking constraint violation will need more degrees of freedom to efficiently reorder the trains because of the greedy and fail-fast principles. Conversely, the MIP-initialiser experiences decreased computation time due to fewer variables that need to be optimised by the MIP solver, resulting in a more time-efficient optimisation. The results show no immediate relation

Table 4
Metrics are averages of 10 runs. "Avg. comp. time" = Average computation time by the GA. "Avg. init. time" = Average initiation time by the MIP-initialiser. Scenarios are defined in Table 3.

| Scenario | Avg. HV | $\sigma_{H V}$ | Avg. $\Delta$ | $\sigma_{\Delta}$ | Avg. comp. time (s) | Avg. init. time (s) | Avg. total time (s) |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 1 | 0.72 | 0.015 | 0.47 | 0.045 | 02.8 | 00.3 |  |
| 2 | 0.69 | 0.007 | 0.53 | 0.031 | 14.8 | 04.2 |  |
| 3 | 0.83 | 0.020 | 0.59 | 0.038 | 13.1 | 05.1 |  |
| 4 | 0.69 | 0.010 | 0.63 | 0.043 | 22.6 | 09.2 | 19.0 |
| 5 | 0.81 | 0.015 | 0.75 | 0.101 | 21.4 | 10.6 | 18.2 |
| 6 | 0.83 | 0.008 | 0.58 | 0.041 | 20.0 | 31.8 |  |
| 7 | 0.71 | 0.009 | 0.76 | 0.047 | 31.5 | 15.7 |  |
| 8 | 0.80 | 0.016 | 0.67 | 0.038 | 30.6 | 17.8 | 32.0 |
| 9 | 0.83 | 0.014 | 0.63 | 0.038 | 28.5 | 20.5 | 32.5 |
| 10 | 0.76 | 0.009 | 0.49 | 0.065 | 28.9 | 23.7 | 48.2 |

between network complexity and diversity. However, better diversity was achieved in scenarios $1,2,3,6$, and 10 . For the worst-case scenario, the average total computation time was less than one minute: 52.6 s , which is reasonable for usage in a decision support tool. However, more complex networks that are not linear may affect the computation time.

Additionally, Table 4 can be used to analyse the performance and improvements made to the Pareto front by the GA compared to the MIP-initialiser that is solving the same model by different approaches. From Table 4 it is evident that the MIP-initialiser is outperforming the GA based on the average running time. However, if comparing the quality of the solutions, Fig. 12(a) shows that it was only possible to achieve $H V=0.65$ and $\Delta=0.75$ with the MIP-initialiser. Whereas Table 4 shows that the GA improves these metrics up to $H V=0.80$ and down to $\Delta=0.67$ for scenario 8 which is the most comparable to the scenario used in Fig. 12(a). This shows that the GA was able to improve the solutions by $6.66 \%$ in HV and $10.66 \%$ in $\Delta$. These results are expected since the MIP-initialiser is not constructed to produce as diverse Pareto-optimal solutions, but rather to produce feasible solutions to be diversified by the GA.

### 5.4. Timetable optimisation analysis

An important aspect of the results is relating it to the business case by evaluating how well the approach performs measured by the objectives (i.e. the business KPIs). First, a comparison of an optimised timetable against the original timetable in service is provided to reflect on how the planned run time affects the objectives. Next, five Pareto-optimal timetables are given to show the diversity of the solutions. Last, the results are reflected in terms of return in profit.

Table 5 compares an original timetable in service from ML-1 to SW-27 against a Pareto-optimal timetable. This example uses all 18 timetables unlocked on the entire network to achieve maximal energy-saving potential. By extending the timetable by 11 min and 40 s , the energy consumption can be decreased by $4 \%$, however with the cost of increasing the passenger travel time by $2.51 \%$. Looking at the sections, there seems to be a high potential for saving energy from SW-23 to SW-24 and from SW-24 to SW-26 with savings of more than $20 \%$ by adding one minute of extra slack time. The sections from SW- 3 to SW-7 and SW-7 to SW-10 have a difference in energy consumption of $0 \%$ because both running times are after the global minimum, meaning the energy consumption will be constant. Though, this is possibly due to a misfit of the energy functions. The section from SW-26 to SW-27 also has a difference in energy consumption of $0 \%$, however, this is caused by it was not possible to fit an energy function to the underlying data due to missing data or too many outliers. The running time was therefore locked. Note that not all timetables in the network will show similar performance. Some timetables will receive more slack time and others less to let timetables with a higher saving potential gain bigger absolute savings. This is to minimise the objectives over the sum of the timetables.

Although most of the relevant papers found in the literature review could achieve a higher reduction in energy consumption, our results cannot be compared. Previous work either relied on simulated data or optimal speed profiles to determine the energy consumption of the optimised timetables. On the other hand, our results are based on extensive historical data from the train operation of an entire railway network used to estimate the actual energy consumption. To our knowledge, we are the first to contribute such results in TTP.

Table 5 only shows a single timetable out of the 18 optimised timetables. Instead, we turn our attention towards the network solutions to see the diversity of the solutions. Table 6 compares five Pareto-optimal timetable networks for different objectives. Overall, our approach can save up to $4.83 \%$ energy and decrease the passenger travel time by at most $8.72 \%$ compared to the original timetable in service. If the Pareto-optimal solutions are fixed to the current energy consumption, then the passenger travel time can be decreased by $4.64 \%$, and for the opposite, the energy consumption can be decreased by $3.3 \%$. Last, the median solution (i.e. equally weighting each objective) can save $2.03 \%$ in energy consumption and $2.9 \%$ in passenger travel time. Fig. 14 shows these five solutions in the context of the full Pareto front and the original timetable. The grey lines indicate the objective values of the original timetable. Ideally, the timetable planner will select or adjust a solution within the grey lines to avoid compromising one objective over the other.

These results can be related to the return in profit for a specific TOC. The Danish State Railways (DSB) will be used as an example because they also run with GreenSpeed from Cubris (Andersen, 2018). In 2019, they consumed 249,623 MWh electricity and 63.7 million litres of diesel (DSB, 2020). Assuming our approach on average will save $3 \%$ in energy consumption, this will equal a saving

Table 5
Comparison of an optimised timetable against the original timetable in service.

| Origin | Destination | Original <br> run time hh:mm:ss | Optimised <br> run time <br> hh:mm:ss | Difference |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  | Run time $\pm \mathrm{mm}$ :ss | EC \%Wh/t |
| ML-1 | ML-4 | 00:07:30 | 00:06:47 | -00:43 | +00.19\% |
| ML-4 | ML-18 | 00:17:00 | 00:18:40 | +01:40 | -07.93\% |
| ML-18 | ML-26 | 00:18:30 | 00:20:23 | +01:53 | -03.20\% |
| ML-26 | SW-3 | 00:15:30 | 00:16:14 | +00:44 | -02.89\% |
| SW-3 | SW-7 | 00:18:00 | 00:19:54 | +01:54 | $\pm 00.00 \%$ |
| SW-7 | SW-10 | 00:18:30 | 00:16:02 | -02:28 | $\pm 00.00 \%$ |
| SW-10 | SW-11 | 00:09:30 | 00:09:22 | -00:08 | +00.52\% |
| SW-11 | SW-12 | 00:07:00 | 00:08:56 | +01:56 | -07.15\% |
| SW-12 | SW-13 | 00:06:30 | 00:07:22 | +00:52 | -11.61\% |
| SW-13 | SW-14 | 00:05:00 | 00:05:40 | +00:40 | -05.82\% |
| SW-14 | SW-15 | 00:08:30 | 00:09:21 | +00:51 | -02.53\% |
| SW-15 | SW-18 | 00:12:30 | 00:12:43 | +00:13 | -00.45\% |
| SW-18 | SW-20 | 00:11:00 | 00:11:33 | +00:33 | -00.45\% |
| SW-20 | SW-21 | 00:05:00 | 00:05:40 | +00:40 | -03.67\% |
| SW-21 | SW-23 | 00:07:00 | 00:07:09 | +00:09 | -00.68\% |
| SW-23 | SW-24 | 00:03:30 | 00:04:41 | +01:11 | -20.44\% |
| SW-24 | SW-26 | 00:04:00 | 00:05:43 | +01:43 | -28.71\% |
| SW-26 | SW-27 | 00:03:00 | 00:03:00 | $\pm 00: 00$ | $\pm 00.00 \%$ |
| Total |  | 02:57:30 | 03:09:10 | +11:40 | -04.04\% |

Table 6
Comparison between five Pareto-optimal timetable networks.

| Label | Objective | EC <br> Wh/t | PTT <br> s | Diff. EC <br> $\% W h / t$ | Diff. PTT <br> $\% s$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| A | Original | 50122.40 | 18570280.04 | $\pm 0.00 \%$ | $\pm 0.00 \%$ |
| B | Lowest PTT | 54493.45 | 16902395.15 | $+8.72 \%$ | $-8.98 \%$ |
| C | Fixed EC | 50121.59 | 17707819.79 | $-0.01 \%$ | $-4.64 \%$ |
| D | Median EC \& PTT | 49105.72 | 18031787.76 | $-2.03 \%$ | $-2.90 \%$ |
| E | Fixed PTT | 48467.92 | 18381084.43 | $-3.30 \%$ | $-1.02 \%$ |
| F | Lowest EC | 47702.28 | 19712213.96 | $-4.83 \%$ | $+6.15 \%$ |



Fig. 14. The full Pareto front of the timetables shown in Table 6.
of 7488 MWh and 1.91 million litres of diesel per year. Thus, assuming a price of around $€ 0.3$ per kWh (Eurostat, 2021) and $€ 1.4$ per litre diesel (Fuels Europe, 2019) (2019 Denmark prices), the total financial saving for DSB would be more than €4,900,000 each year. Thus, applying EETT in a TOC can result in significant savings in energy consumption and will be an attractive opportunity. However, applications in larger and more complex networks with more constrained adjustments may yield lower savings.

## 6. Conclusion and future work

Railway transportation is one of the most energy-efficient modes of land transportation and becomes increasingly important to alleviate the effects of climate change. Energy-efficient train timetables can help TOCs cut costs and contribute towards a reduction in $\mathrm{CO}_{2}$ emissions.

This article presented a data-driven approach to the multi-objective network-optimised energy-efficient train time-tabling problem. The problem formulation is based on a macroscopic network model and considers constraints for run time, dwell time, section and station capacities, overtaking, and locking timetable times. The objectives taken into account were passenger travel time and energy consumption. To our knowledge, we are the first to use extensive historical data from the train operation of an entire railway network to model the actual energy consumption. Using this kind of data for optimising railway timetables is a new contribution to the research. The proposed algorithm was a bi-objective matheuristic using an NSGA-II-based GA to generate a set of Pareto-optimal solutions. The feasibility of solutions throughout the algorithm's running time was ensured by a greedy FCFS fail-fast repair heuristic applied after the mutation. A MIP-initialiser was introduced to generate the initial population with satisfactory diversity and quality of solutions.

The used crossover operator is a hybrid between the single-point and average crossovers, which showed promising results in our experimental results compared to their performance individually. Given an initial population, the GA produces consistent results with a relatively low standard deviation. The matheuristic was applied to a real-world case from a large North European TOC with a network consisting of 107 stations and 18 timetables. Without sacrificing the other objective, an up to $3.3 \%$ energy saving and $4.64 \%$ reduction in passenger travel time can be achieved, which amounts to over $€ 4,900,000$ per year if applied to the Danish State Railways (DSB), thus, making it attractive for TOCs to look into energy-optimisation of railway timetables as proposed in this article. In the worst case, the running time of the initial population and the GA was in total 52.6 s on average making it suitable for integration with a decision support tool.

In future work, three improvements should be considered. First, further research is needed to explore efficient repair heuristics. In this regard, large neighbourhood search methods may help cope with heavily destroyed solutions needing a non-greedy repair heuristic. Related to this, it is interesting to investigate the impact of more thorough repairs on the $\Delta$ diversity metric of the Pareto front. Second, the algorithm should be extended to work for complex non-linear network structures, tracks with bidirectional traffic, and asymmetric sections and stations with different numbers of tracks in each direction. These extensions allow the algorithm to be generalised to work for all railway networks. Third, consideration of other objectives is needed to ensure the quality of the train service. Such objectives include punctuality, timetable robustness, and passenger waiting time. This is not a trivial task and requires careful attention to the solution quality and computation time. Last, it would be beneficial to perform simulations on the optimised timetables, in order to test the quality of the timetables in terms of robustness and passenger flows. Such simulations could also be used to improve the formulation of the PTT objective by taking into account changes in the passenger flow when arrival and departure times are changed.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The authors do not have permission to share data.

## References

Abd Rahman, R., Ramli, R., 2013. Average concept of crossover operator in real coded genetic algorithm. In: International Proceedings of Economics Development and Research, Vol. 63. IACSIT Press, p. 73.
Andersen, K.H., 2018. Grn blge. Ud Se May 2018, 20.
Arenas, D., Chevrier, R., Hanafi, S., Rodriguez, J., 2015. Solving the Train Timetabling Problem, a mathematical model and a genetic algorithm solution approach. In: 6th International Conference on Railway Operations Modelling and Analysis (RailTokyo2015). Tokyo, Japan, https://hal.archives-ouvertes.fr/hal-01338609. Assad, A., 1980. Models for rail transportation. Transp. Res. A: Gen. 14 (3), 205-220.
Assad, A., 1981. Analytical models in rail transportation: An annotated bibliography. INFOR: Inf. Syst. Oper. Res. 19 (1), 59-80.
Borisovsky, P., Dolgui, A., Eremeev, A., 2009. Genetic algorithms for a supply management problem: MIP-recombination vs greedy decoder. European J. Oper. Res. 195 (3), 770-779. http://dx.doi.org/10.1016/j.ejor.2007.06.060.
Cacchiani, V., Toth, P., 2012. Nominal and robust train timetabling problems. European J. Oper. Res. 219 (3), 727-737. http://dx.doi.org/10.1016/j.ejor.2011. 11.003.

Caimi, G., Kroon, L., Liebchen, C., 2017. Models for railway timetable optimization: Applicability and applications in practice. J. Rail Transp. Plan. Manage. 6 (4), 285-312. http://dx.doi.org/10.1016/j.jrtpm.2016.11.002.

Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and solving the train timetabling problem. Oper. Res. 50 (5), 851-861. http://dx.doi.org/10.1287/opre.50. 5.851.362.

Caprara, A., Kroon, L., Monaci, M., Peeters, M., Toth, P., 2007. Chapter 3 passenger railway optimization. In: Barnhart, C., Laporte, G. (Eds.), Transportation. In: Handbooks in Operations Research and Management Science, vol. 14, North Holland, Amsterdam and London, pp. 129-187. http://dx.doi.org/10.1016/S0927-0507(06)14003-7.
Cordeau, J.-F., Toth, P., Vigo, D., 1998. A survey of optimization models for train routing and scheduling. Transp. Sci. 32 (4), 380-404. http://dx.doi.org/10. 1287/trsc.32.4.380.
2021. Cubris - A. thales company. http://www.cubris.dk/ [Accessed 2021-05-11].

Cucala, A.P., Fernández, A., Sicre, C., Domínguez, M., 2012. Fuzzy optimal schedule of high speed train operation to minimize energy consumption with uncertain delays and driver's behavioral response. Eng. Appl. Artif. Intell. 25 (8), 1548-1557. http://dx.doi.org/10.1016/j.engappai.2012.02.006.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6 (2), $182-197$. http://dx.doi.org/10.1109/4235.996017.
DSB, 2020. Miljopgrelse 2019. https://www.dsb.dk/globalassets/arsrapport/2019/dsb-miljoarsopgorelse-2019.pdf [Accessed 2021-07-04].
Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD '96, AAAI Press, pp. 226-231, URL: http://dl.acm.org/citation.cfm?id=3001460.3001507 [Accessed 2019-06-08].
European Commission, 2019. The European green deal. https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en [Accessed 2021-11-08]. Eurostat, 2021. Electricity prices by type of user. https://ec.europa.eu/eurostat/databrowser/view/ten00117/default/table?lang=en [Accessed 2021-07-04].
Fuels Europe, 2019. Breakdown of automotive diesel prices across EU (March 2019). https://www.fuelseurope.eu/interactive-graph/breakdown-of-automotive-diesel-prices-across-eu-march-2019/ [Accessed 2021-07-04].
Gao, Y., Yang, L., Gao, Z., 2018. Energy consumption and travel time analysis for metro lines with express/local mode. Transp. Res. D 60, 7-27. http: //dx.doi.org/10.1016/j.trd.2016.10.009.
Garrisi, G., Cervelló-Pastor, C., 2019. Train-scheduling optimization model for railway networks with multiplatform stations: Sustainability, $12(1)$, 257. Sustainability 12 (1), 257. http://dx.doi.org/10.3390/SU12010257.
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. B 99, 167-182. http://dx.doi.org/10.1016/j.trb.2016.12.016, URL: https://www.sciencedirect.com/science/article/pii/S0191261516301667.
Haghani, A.E., 1987. Rail freight transportation: A review of recent optimization models for train routing and empty car distribution. J. Adv. Transp. 21 (2), 147-172. http://dx.doi.org/10.1002/atr. 5670210205.
Hansen, I., Pachl, J. (Eds.), 2014. Railway Timetabling and Operations: Analysis, Modelling, Optimisation, Simulation, Performance, Evaluation. Eurail Press.
Harrod, S.S., 2012. A tutorial on fundamental model structures for railway timetable optimization. Surv. Oper. Res. Manag. Sci. 17 (2), 85-96. http: //dx.doi.org/10.1016/j.sorms.2012.08.002.
Higgins, A., Kozan, E., Ferreira, L., 1997. Heuristic techniques for single line train scheduling. J. Heuristics 3 (1), 43-62. http://dx.doi.org/10.1023/A: 1009672832658.

Huang, Y., Yang, L., Tang, T., Gao, Z., Cao, F., 2017. Joint train scheduling optimization with service quality and energy efficiency in urban rail transit networks. Energy 138, 1124-1147. http://dx.doi.org/10.1016/j.energy.2017.07.117.
Huisman, D., Kroon, L.G., Lentink, R.M., Vromans, M.J.C.M., 2005. Operations research in passenger railway transportation. Stat. Neerl. 59 (4), $467-497$. http://dx.doi.org/10.1111/j.1467-9574.2005.00303.x.
Keersmaecker, S.D., Meder, S., 2020. The journey begins - 2021 is the European year of rail!. https://ec.europa.eu/commission/presscorner/detail/en/IP_20_2528 [Accessed 2021-05-12].
Kroon, L.G., Peeters, L.W.P., 2003. A variable trip time model for cyclic railway timetabling. Transp. Sci. 37 (2), 198-212. http://dx.doi.org/10.1287/trsc.37.2. 198.15247.

Lusby, R.M., Larsen, J., Bull, S., 2018. A survey on robustness in railway planning. European J. Oper. Res. 266 (1), 1-15. http://dx.doi.org/10.1016/j.ejor.2017. 07.044.

Lusby, R.M., Larsen, J., Ehrgott, M., Ryan, D., 2011. Railway track allocation: models and methods. OR Spectrum 33 (4), 843-883. http://dx.doi.org/10.1007/ s00291-009-0189-0.
Madsen, M.B., Als, M.V.H., Jensen, R.M., Gram, S.E., 2019. A decision support tool for energy-optimising railway timetables based on behavioural data. In: Paternina-Arboleda, C., Voß, S. (Eds.), Computational Logistics. In: Lecture Notes in Computer Science, vol. 11756, Springer International Publishing, Cham, pp. 397-412. http://dx.doi.org/10.1007/978-3-030-31140-7_25.
Mitchell, M., 1998. An Introduction to Genetic Algorithms. MIT Press.
Nachtigall, K., Voget, S., 1997. Minimizing waiting times in integrated fixed interval timetables by upgrading railway tracks. European J. Oper. Res. 103 (3), 610-627. http://dx.doi.org/10.1016/S0377-2217(96)00284-6, URL: http://www.sciencedirect.com/science/article/pii/S0377221796002846.
Scheepmaker, G., Goverde, R.M., 2015. Running time supplements: Energy-efficient train control versus robust timetables. In: Tomii, N., Hansen, I., Hirai, C. (Eds.), Proceedings of the 6th International Conference on Railway Operations Modelling and Analysis, RailTokyo2015. International Association of Railway Operations Research, pp. 1-20.
Scheepmaker, G.M., Goverde, R.M., Kroon, L.G., 2017. Review of energy-efficient train control and timetabling. European J. Oper. Res. 257 (2), $355-376$. http://dx.doi.org/10.1016/j.ejor.2016.09.044.
Scheepmaker, G.M., Pudney, P.J., Albrecht, A.R., Goverde, R.M., Howlett, P.G., 2020. Optimal running time supplement distribution in train schedules for energy-efficient train control. J. Rail Transp. Plan. Manage. 14, 100180. http://dx.doi.org/10.1016/j.jrtpm.2020.100180.
Semet, Y., Schoenauer, M., 2005. An efficient memetic, permutation-based evolutionary algorithm for real-world train timetabling. In: The 2005 IEEE Congress on Evolutionary Computation. IEEE, Piscataway, pp. 2752-2759. http://dx.doi.org/10.1109/CEC.2005.1555040.
Serafini, P., Ukovich, W., 1989. A mathematical model for periodic scheduling problems. SIAM J. Discrete Math. 2 (4), 550-581. http://dx.doi.org/10.1137/ 0402049.

Sicre, C., Cucala, P., Fernández, A., Jiménez, J.A., Ribera, I., Serrano, A., 2010. A method to optimise train energy consumption combining manual energy efficient driving and scheduling. In: Ning, B., Brebbia, C.A. (Eds.), Computers in Railways XII. In: WIT Transactions on The Built Environment, WIT PressSouthampton, UK, pp. 549-560. http://dx.doi.org/10.2495/CR100511.
Su, S., Li, X., Tang, T., Gao, Z., 2013. A subway train timetable optimization approach based on energy-efficient operation strategy. IEEE Trans. Intell. Transp. Syst. 14 (2), 883-893. http://dx.doi.org/10.1109/TITS.2013.2244885.
Su, S., Tang, T., Li, X., Gao, Z., 2014. Optimization of multitrain operations in a subway system. IEEE Trans. Intell. Transp. Syst. 15 (2), $673-684$. http://dx.doi.org/10.1109/TITS.2013.2285737.
Tang, L., D'Ariano, A., Xu, X., Li, Y., Ding, X., Samà, M., 2021. Scheduling local and express trains in suburban rail transit lines: Mixed-integer nonlinear programming and adaptive genetic algorithm. Comput. Oper. Res. 135, 105436. http://dx.doi.org/10.1016/j.cor.2021.105436.
Tormos, P., Lova, A., Barber, F., Ingolotti, L., Abril, M., Salido, M.A., 2008. A genetic algorithm for railway scheduling problems. In: Kacprzyk, J., Xhafa, F., Abraham, A. (Eds.), Metaheuristics for Scheduling in Industrial and Manufacturing Applications. In: Studies in Computational Intelligence, vol. 128, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 255-276. http://dx.doi.org/10.1007/978-3-540-78985-7_10.
Wang, M., Wang, L., Xu, X., Qin, Y., Qin, L., 2019. Genetic algorithm-based particle swarm optimization approach to reschedule high-speed railway timetables: A case study in China. J. Adv. Transp. 2019, 1-12. http://dx.doi.org/10.1155/2019/6090742.
Xu, X., Li, K., Li, X., 2016. A multi-objective subway timetable optimization approach with minimum passenger time and energy consumption. J. Adv. Transp. 50 (1), 69-95. http://dx.doi.org/10.1002/atr. 1317.
Yang, X., Chen, A., Li, X., Ning, B., Tang, T., 2015. An energy-efficient scheduling approach to improve the utilization of regenerative energy for metro systems. Transp. Res. C 57, 13-29. http://dx.doi.org/10.1016/j.trc.2015.05.002.

Yang, X., Li, X., Ning, B., Tang, T., 2016. A survey on energy-efficient train operation for urban rail transit. IEEE Trans. Intell. Transp. Syst. 17 (1), $2-13$. http://dx.doi.org/10.1109/TITS.2015.2447507.
Yang, X., Wu, J., Sun, H., Gao, Z., Yin, H., Qu, Y., 2019. Performance improvement of energy consumption, passenger time and robustness in metro systems: A multi-objective timetable optimization approach. Comput. Ind. Eng. 137, 106076. http://dx.doi.org/10.1016/j.cie.2019.106076.
Yin, J., Yang, L., Tang, T., Gao, Z., Ran, B., 2017. Dynamic passenger demand oriented metro train scheduling with energy-efficiency and waiting time minimization: Mixed-integer linear programming approaches. Transp. Res. B 97, 182-213. http://dx.doi.org/10.1016/j.trb.2017.01.001.
Yu, X., Lu, Y., Yu, X., 2018. Evaluating multiobjective evolutionary algorithms using MCDM methods. Math. Probl. Eng. 2018.


[^0]:    This article is researched in collaboration with Cubris.

    * Corresponding author.

    E-mail addresses: matt@itu.dk (M.V.H. Als), mabm@itu.dk (M.B. Madsen), rmj@itu.dk (R.M. Jensen).
    https://doi.org/10.1016/j.jrtpm.2023.100374
    Received 23 May 2022; Received in revised form 3 January 2023; Accepted 1 March 2023
    Available online 28 March 2023
    2210-9706/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

