A decision support tool for energy-optimising railway timetables based on behavioural data*

Mathias Bejlegaard Madsen[1], Matthias Villads Hinsch Als[1], Rune M. Jensen[1], and Sune Edinger Gram[2]

[1]IT University of Copenhagen, Rued Langaards Vej 7, 2300 Copenhagen S, Denmark
{mabm,matt,rmj}@itu.dk
[2]Cubris - A Thales Company, Mileparken 22, 2740 Skovlunde, Denmark
s.gram@cubris.dk

Abstract. Energy-efficient train operation can reduce operating costs and contribute to a reduction in CO_2 emissions. To utilise the full potential of energy-efficient driving, energy-efficient timetabling is crucial. To address this problem, we propose a decision support tool to give timetable planners insight into energy consumption for a given timetable. The decision support tool uses a recommendation based on quadratic optimisation of a given timetable. Differently to previous work, the optimisation uses actual data from the train operation, which is pre-processed by data reduction, outlier detection, and second-degree regression modelling. With this approach, our results show that the optimised timetables can save up to 33.07% energy on a single section and up to 6.23% for a complete timetable. Solutions are computed in less than a microsecond.

Keywords: Energy-efficient train timetabling · Decision support · Data mining · Quadratic optimisation

1 Introduction

The threat of climate change urges for energy-efficient solutions to the transportation sector. Even though railway provides one of the most energy-efficient forms of transportation, there is still significant potential for reduction in energy consumption [2]. Energy-efficient train operation is one approach that both can reduce operating costs and contribute to a reduction in CO_2 emissions [8]. To utilise the full potential of energy-efficient driving, energy-efficient timetabling is crucial. Timetable planners are experts in the complex process of railway timetabling, which has to conform to several known internal and external factors. What timetable planners seldom know is how changes to a timetable affect the total energy consumption. For that reason, insights into minimisation of energy consumption in railway timetables are a key aspect in reducing CO_2 emissions, while sustaining a business opportunity of the consequently reduced costs.

* This research is conducted in collaboration with Cubris - A Thales Company.

The problem has attracted attention in recent years. C. Sicre et al. [10] uses Pareto frontiers for optimisation to redistribute slack time of a journey. S.D. Gupta et al. [7] uses a linear programming optimisation model to optimise energy consumption. The approach takes many constraints into account, including trip time, dwell time, headway, cross-over, connection, and total time. The model is applied to optimise a metro spanning a full-service period of one day. The worst case optimisation gave a 19.27% improvement, while the best case gave a 21.61% improvement. A.P. Cucala et al. [5] suggests a fuzzy linear programming model taking the driver's behavioural response into account. The paper furthermore models uncertainty in delays as fuzzy numbers. The approach was tested on a high-speed line in Spain, which achieved a decrease of 6.7% in energy consumption compared to the timetable in service. Although all aforementioned papers achieved acceptable results in decrease of energy consumption, the methods was modelled using simulated data. As such, the data does not reflect real world restrictions and conditions of a running railway network. This will inevitably create deviating predictions of energy consumption compared to actual measurements, making it insufficient for a decision support tool.

G.M. Scheepmaker et al. [9] distributes the available slack time uniformly. The paper is based on timtables from the Dutch railway and achieved a 7.2% decrease in energy consumption. In practice, an uniform distribution would only be sufficient in situations where enough slack time is available. With less slack time available, the paper will achieve worse results than papers [10, 7, 5], since it does not prioritise the distribution to minimise energy consumption.

This paper proposes a decision support tool to give timetable planners insight into energy consumption for a given timetable to address the problem mentioned above. To our knowledge, we contribute the first end-to-end analysis of real data from rolling stock and show how it can be used to provide an optimised timetable based on the train drivers' actual behaviour. The result is an interactive and intuitive decision support tool letting the timetable planner adjust the optimised timetable according to their domain knowledge and experience.

The remainder of this paper is structured as follows. Section 2 formulates the problem and Section 3 proposes an approach to solve it. Section 4 evaluates the approach and its design choices. Last, Section 5 concludes and discusses future work.

2 Problem Formulation

Railway timetable planning is a complex process that includes several hard and soft constraints. The hard constraints such as network capacity, rolling stock availability and headway are known by the timetable planner, as opposed to the soft constraints such as punctuality and energy consumption. Though, it is infeasible to include all constraints to the timetable-optimisation. Train timetabling, in general, is NP-Hard [3] due to the combination of several constraints and requirements. To reduce complexity, one compromise is to consider the soft constraints in a decision support tool and let the timetable planner make valid adjustments according to the hard constraints. The estimation of soft constraints can be approached either theoretically or operationally. The latter is the focus of this paper and requires data from train operation.

To this end, we have collaborated with Cubris [4]. Cubris develops a driver advisory system named GreenSpeed. GreenSpeed improves punctuality and reduces energy consumption. GreenSpeed has been deployed for multiple train operating companies. Regarding optimisation of their customers' operation, Cubris can with permission gain access to massive quantities of data produced by GreenSpeed over the years. Data based on more than 140,000 train runs were accessible for this paper. This data contains information about how train operators have been driving the trains. The data includes precise energy consumption and exact arrival and departure times as shown in Figure 1(a). For this paper, we assume that the trains already are equipped with GreenSpeed. All data from Cubris is normalised and anonymised.

As mentioned in the introduction, the key idea is to give insights into the energy consumption of a timetable. Based on actual data, an intuitive decision support tool is proposed to give timetable planners insight into the energy efficiency of current timetables. Comparing a timetable to another is one of the approaches available when deciding how a timetable should be adjusted. To give the user an intelligent way of evaluating the energy efficiency of a timetable, a recommendation of the optimal timetable can be visualised, while enabling the user to adjust the timetable to match hard constraints. The user's adjustments will be restricted within the range of actual data to provide the highest accuracy of actual energy consumption. The recommendation is limited to redistribute the slack time given in the timetable only. The decision support tool only modifies one timetable at once with the data that identical timetables can provide.

3 Solution Approach

We propose a five-step approach to the solution: 1) data reduction, 2) outlier detection, 3) regression modelling, 4) optimisation of energy functions, and 5) decision support tool. We describe these steps in the following sections.

3.1 Data Reduction

To be able to predict the energy consumption with high accuracy, the data has to be pre-processed. The pre-processing removes invalid data such as sections with no actual energy consumption, arrival and departure times, and train formation specified. Then the data is reduced to a subset containing only the relevant features which have an impact on energy consumption. A plot showing a section with no reduction is omitted due to the high number of data points.

First, the planned section run time affects the slope of the energy function, and the distribution of points being relative to the planned arrival time. The planned run time is the time between departure at the previous station and arrival at the next station. Figure 1(a) shows the result of only looking at one section run time. A detailed comparison and argumentation for this choice can be found in Section 4.1. Second, different train formations consume different amounts of energy based on the specifications of the train[1] and how many trains and carriages are linked together. As seen in Figure 1(a), the data points are clustered in three groups, each representing different train formations. Figure 1(b) shows the same plot after section run time and train formation has been constrained to only include identical values respectively. It should be noted, that due to a high number of data points, the figure shown has been sampled to fewer points. Third, our experiments in Section 4.1 show that outliers often are a result of drivers with a high accumulated delay. Based on the experiments, an interval of ±150 seconds from the planned arrival time was introduced. Figure 1(c) shows the impact of removing highly delayed train runs. Last, to avoid having multiple energy consumptions per run time, the run times are aggregated and the energy consumption is averaged for each run time. Figure 1(d) shows the final result.

A trend is now clearly visible in the plotted data. Not surprisingly, the trend shows that if the train driver has less time to drive the section, the train will consume more energy. To further improve the accuracy and quality of the data, additional methods for outlier detection will be discussed next.

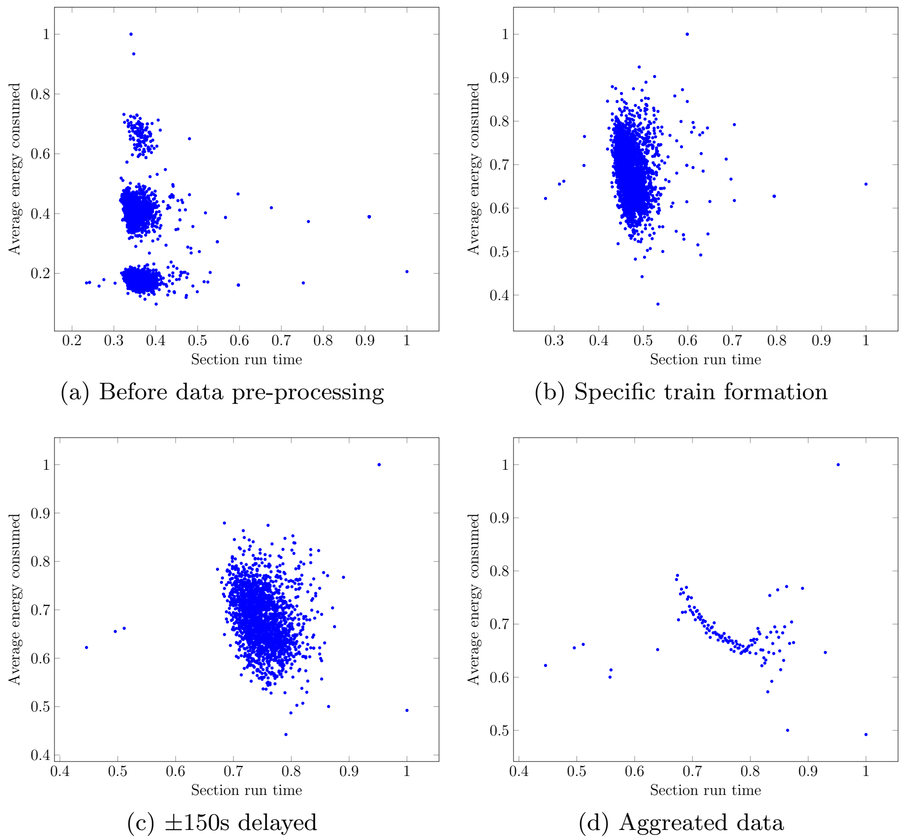
3.2 Outlier Detection

In Section 3.1, we discarded points that did not belong to the context of interest by setting constraints on the values. The objective of this section is to remove data points which do not conform to the trend. In other words, the outliers are removed. Outliers in this domain include extraordinary driving conditions. This includes bad weather, malfunctions and signal errors. If not removed, the outliers reduce data quality and lower prediction accuracy. We use unsupervised machine learning techniques for outlier detection. Especially density-based techniques are relevant because of the dense main cluster as shown in Figure 2. This completely disqualifies widely used centroid-based clustering algorithms. Therefore, the density-based clustering algorithm DBSCAN [6] was chosen for the solution. DBSCAN's robustness to noise makes it a good fit. DBSCAN can use several distance measures; in this work, the Euclidean distance is used.

The data for each section will vary when it comes to the distance between points which roots in the section run properties, i.e. the section to be run and the train formation. Normalisation even out a lot of the density variations, but extreme outliers will also affect the perceived density of the main cluster if not caught by the reduction in Section 3.1. When the data has been clustered, noise and all clusters not being the main cluster are removed.

Figure 2 shows an example of detected outliers with DBSCAN. However, DBSCAN only produces a good output if the parameters ε and $MinPts$ are chosen properly. The ε-parameter is the maximum distance between two points to be assigned to the same cluster, and Section 4.2 describes how we arrived at $\varepsilon = 0.08$ as a value that provide satisfactory result. The reader should note that

[1] Examples are engine type, length, and weight.



(a) Before data pre-processing

(b) Specific train formation

(c) ±150s delayed

(d) Aggreated data

Fig. 1. Data reduction process.

a satisfactory outlier removal does not mean a perfect outlier removal, as the latter cannot be defined properly.

The other parameter to be chosen is $MinPts$. $MinPts$ is the minimum number of data points to form a cluster. This parameter is less sensitive in this use case, though, it cannot be too large. If the parameter value is too large, many points will be considered noise and will be discarded, because there are not enough data points to form the main cluster in low-frequency data sets. If the parameter value is too low, DBSCAN might make a lot smaller local clusters alongside the main cluster. Setting it too low, will not affect the main cluster at all, and the small clusters are filtered out anyway. Therefore, the $MinPts$-value is set to be 10.

3.3 Regression Modelling

Our goal is to predict the energy consumption for a given amount of time spent running a section. We use regression for this purpose. A regression model is

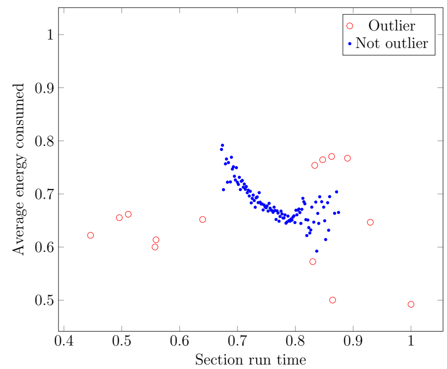


Fig. 2. DBSCAN, $\varepsilon = 0.08$, $MinPts = 10$.

calculated for each section on a journey such that it can be used for optimising the time spent with regards to energy consumption in Section 3.4.

By visual inspection of Figure 1(d), the formation of the points looks like it follows a trend of a second-degree polynomial. This makes good sense, if we consider the formula for kinetic energy in classical mechanics:

$$E_k = \frac{1}{2}mv^2, \tag{1}$$

where m is the mass, v is the velocity and E_k is the kinetic energy. If the train driver has less time to drive the section, they will have to drive with an increased velocity to keep up with the timetable. Hence, the negative change in time to drive the section will increase the consumed energy quadratically. Figure 3 shows second-degree polynomials fitted to four different datasets.

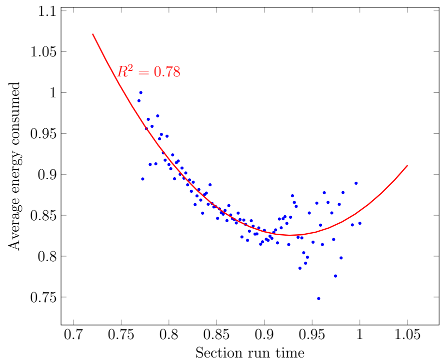
Figure 3 shows that the regression model will predict a rise in energy consumption after the vertex. Based on the formula for kinetic energy, a rise in energy consumption is only expected when the velocity is negative. For this reason, the energy function should be discarded after the vertex.

3.4 Optimisation of Energy Functions

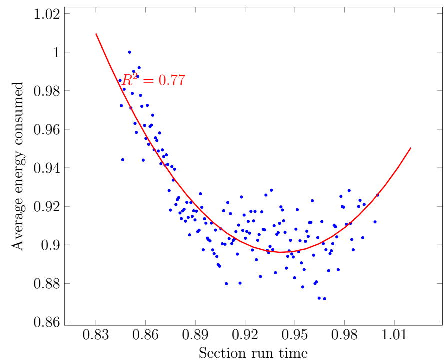
From regression, an energy function was obtained for each section in a timetable. Each section is denoted by $s_1, s_2, ...s_n$, where n is the number of sections contained in a timetable. The energy function for each section is denoted by $e_1(t_1)$, $e_2(t_2), ..., e_n(t_n)$, where t_n is the section run time.

Objective function Given the energy function for each section, the objective function is the sum of energy functions describing the total energy consumption of the timetable, denoted E:

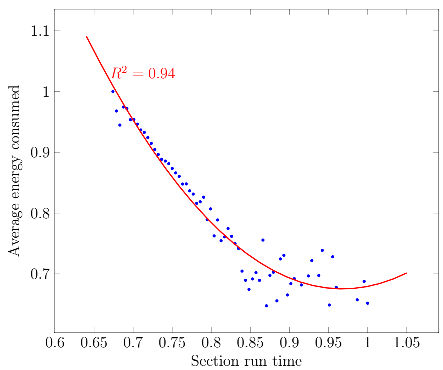
$$E(t) = \sum_{i=1}^{n} e_i(t_i) \tag{2}$$



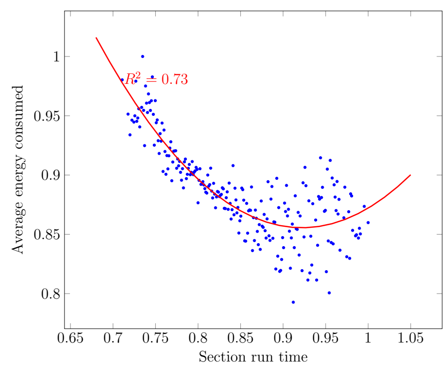
(a)



(b)



(c)



(d)

Fig. 3. Average energy consumption and run time for section runs without outliers, fitted with a second-degree polynomial. Example with four different data sets.

Constraints From the domain described in Section 2, the constraints (3b), (3c), and (3d) have been identified. Allowing the section run times to be increased until the minimum has been reached for all $e_i(t_i)$ in E will result in the best-optimised timetable. However, the cost of doing this is a new timetable which total time of the journey is increased. Equation (3b) ensures that the total run time is equal to the current timetable. Equation (3c) constraints each section by a minimum run time, since physical limitations of the train and tracks restricts the speed of the train. Equation (3d) constraints each section by a maximum run time, since the optimisation variable cannot be greater than the vertex of the quadratic energy function, as discussed in Section 3.3.

To output an energy-efficient timetable, the sum of energy functions for the timetable described by Equation 2, is sought to be minimised. Hence, the optimisation problem can be established as follows:

$$\textbf{minimise } E(t) \tag{3a}$$

$$\textbf{subject to } \sum_{i=1}^{n} t_i = \sum_{i=1}^{n} C_i \tag{3b}$$

$$t_{i_{min}} \leq t_i, \qquad 0 < i \leq n \tag{3c}$$

$$e_i'(t_i) \leq 0, \qquad 0 < i \leq n, \tag{3d}$$

where t is the vector of decision variables $t = (t_1, t_2, ..., t_n)$, n is the number of sections in the timetable, and C is the current timetable containing the run time of each section.

Since $E(t)$ is a sum of convex quadratic functions, the optimisation problem (3a) is a convex quadratic programming problem that can be solved to optimality by a standard commercial solver such as CPLEX.

3.5 Decision Support Tool

The purpose of this section is to present a user interface (UI) which supports timetable planners in making energy-efficient decisions. This is achieved by visualising the cost in energy consumption of adjusting the slack time.

The central part is the timetable configurator, located in the middle of the screen shown in Figure 4. It shows for each section on the journey two sliders; the lower one with two handles to adjust the departure time and arrival time, and the upper one is fixed and shows the original configuration of the timetable. Initially, the lower slider is configured to show the energy-optimised timetable. When the timetable planner adjusts the energy-optimised timetable with their domain knowledge, the changes are reflected in the estimated difference in energy consumption of the section, which is shown just above the sliders. The estimated energy consumption for the whole journey is shown in the upper right portion of the screen. This enables the timetable planner to use energy consumption as a measure of the overall performance of the timetable, by studying how sections contribute differently to the total energy consumption.

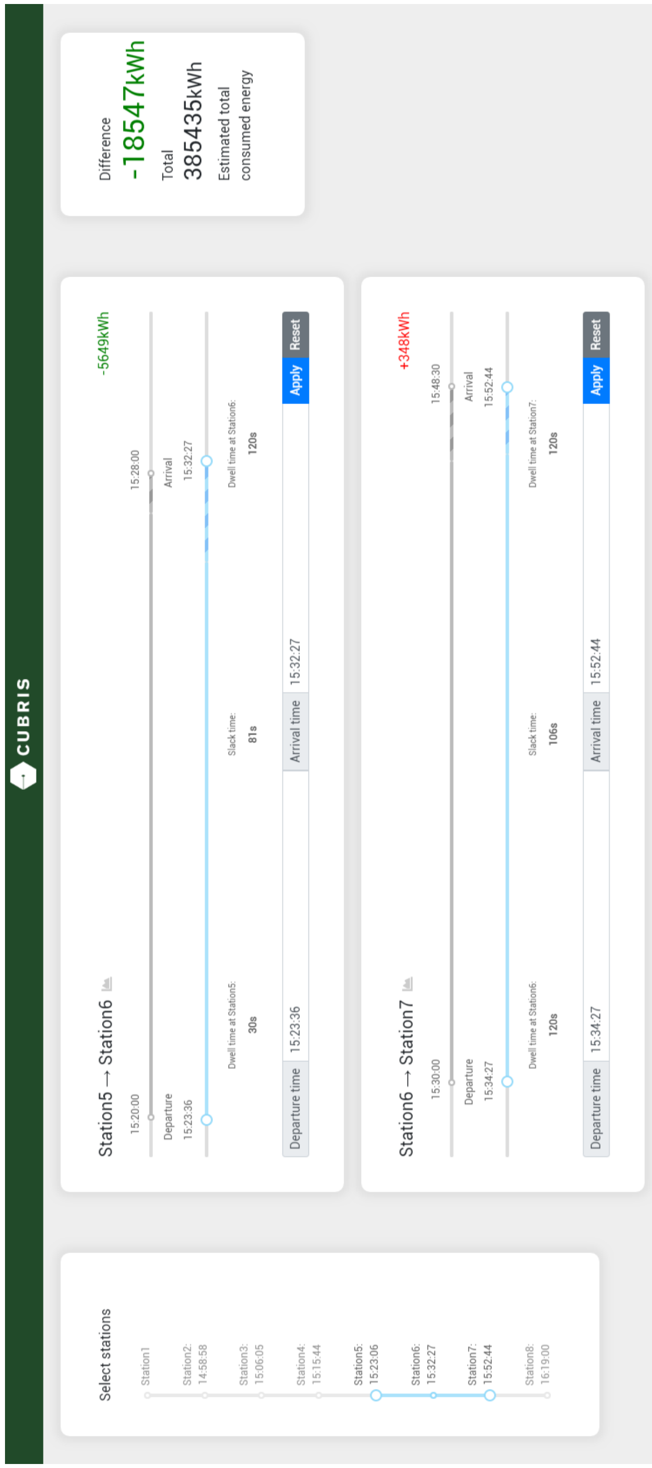


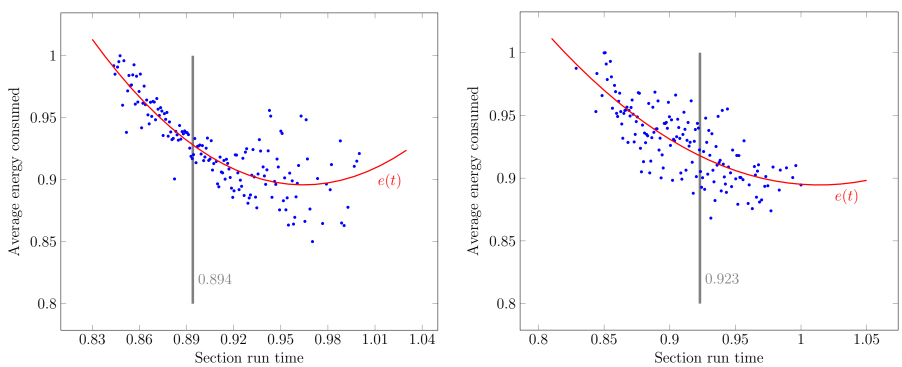
Fig. 4. View for configuring the timetable.

4 Experimental Results

The purpose of our experiments is to evaluate each step conducted in the data reduction and the effect of the parameters for outlier detection. In addition, optimised timetables are compared to the original timetables to evaluate the distribution of the slack time and the energy reduction. The experiments are based on data from actual train runs conducted in a one year period.

4.1 Data reduction

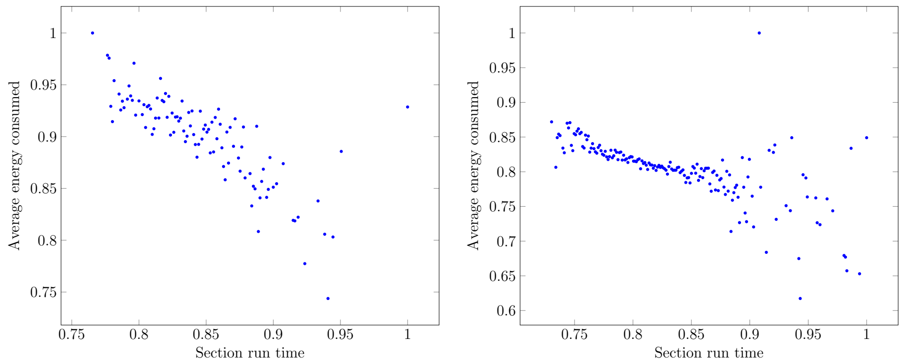
Section run time A section can be a part of multiple timetables with differing planned section run time. Figure 5 shows how two different section run times of the same section effects the slope of the energy function. The same figure shows that the slope of the energy functions is different. The separation of planned run times is therefore important since the slope of the energy function determines the cost of changing the run time. Another issue is that the distribution of points is relative to the planned run time. However, this is not visible in Figure 5 probably due to the difference between the planned run times is too small to have an effect.



(a) Timetable run time = 0.894 (b) Timetable run time = 0.923

Fig. 5. Different run times affecting the energy consumption of section runs between two stations.

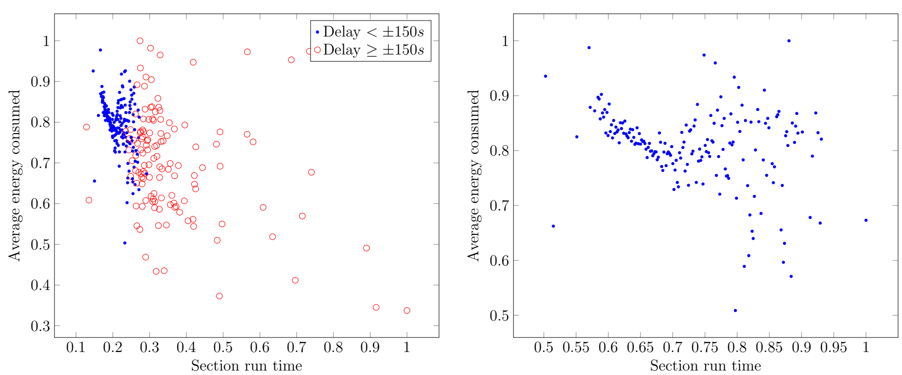
Train formation The length and weight of the train depends on the number and type of carriages. Different formations of trains thereby consume different amounts of energy. Figure 6(a) and 6(b) show typical examples of the energy consumption of two different train formations. From the figures it is noticed that the train formation in Figure 6(a) to the left has a steeper energy consumption compared to the train formation in Figure 6(b). It impacts energy consumption especially if an early arrival time is chosen for Figure 6(a). The cost of choosing an arrival time for Figure 6(b) is not as high. The separation of train formation is therefore important, even though both train formations are in the same range of energy consumption.



(a) Train formation A (b) Train formation B

Fig. 6. Different train formations affecting the energy consumption for section runs between two stations.

Level of delay The delay constraint is chosen to be ±150s to remove outliers. This choice is based on an evaluation of the following delays: 30, 90, 150, 210, and 270 seconds for five different sections. Figure 7(a) shows a plot for an example with all levels of delay. The figure shows that a large portion of outliers are to be removed, while Figure 7(b) shows that the energy trend is preserved after the removal.



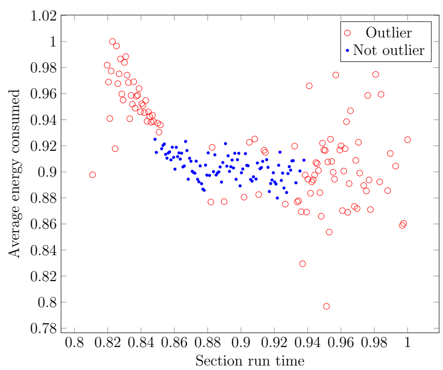
(a) Before removal of delayed trains (b) After removal of delayed trains ($\geq \pm 150s$)

Fig. 7. Level of delay affecting the energy consumption for section runs between two stations.

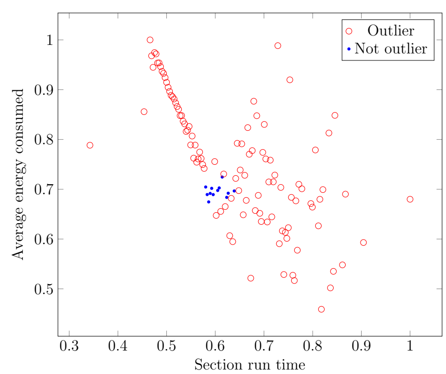
4.2 Parameter-tuning of ε in DBSCAN

Four ε-values were chosen: 0.05, 0.08, 0.1 and 0.15. Each value was applied to five data sets extracted from different sections with varying properties, including train formation, data frequency, the visual shape of the data, planned run time, and the density. A total 20 plots were inspected. If DBSCAN found more than one cluster, the biggest cluster was marked as "not outlier" and everything else as "outlier". The data was outlier detected after data reduction and aggregation.

The value 0.05 marks too many points as outliers in at least two obvious cases as shown in Figure 8(a) and 8(b). In Figure 8(a), an important part of the trend on the upper left side was marked as outliers. In Figure 8(b) only a small fraction of the original data is remaining.



(a) $\varepsilon = 0.05$, $MinPts = 10$

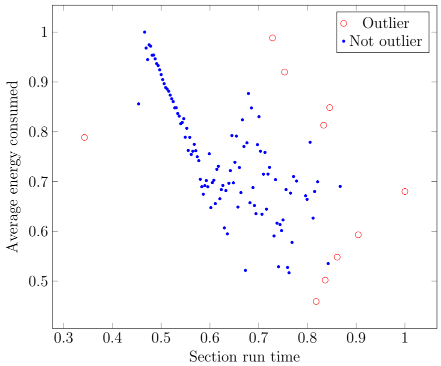


(b) $\varepsilon = 0.05$, $MinPts = 10$

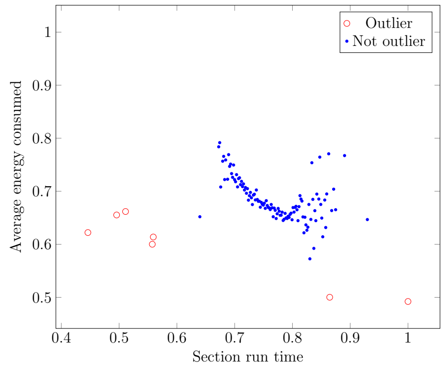
Fig. 8. Comparison of $\varepsilon = 0.05$ on two different data sets

On the other hand, the ε-value 0.15 barely removes any outliers, as shown in Figure 9(a) and 9(b). Here, it is evident that the sparse area around the centre of Figure 9(a) should have been reduced a bit more to avoid noise. Figure 9(b) shows outliers around $(0.7, 0.2)$ and $(0.75, 0.55)$ that would interfere too much with further data analysis.

Based on Figure 10, the ε-parameter is chosen to be 0.08, as it removed a satisfactory amount of noise compared to $\varepsilon = 0.1$. The noise is mainly generated by the behavioural response of the train driver using too much time on the section, consuming more energy to keep up with the timetable. That also proves the limitation of the solution; it should only be used for smaller changes in the timetables as bigger changes will lead to more significant deviations in the prediction due to the behavioural noise.

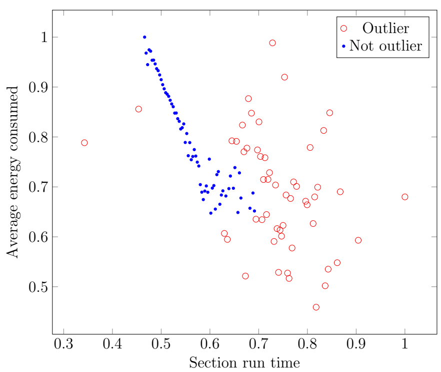


(a) $\varepsilon = 0.15$, $MinPts = 10$

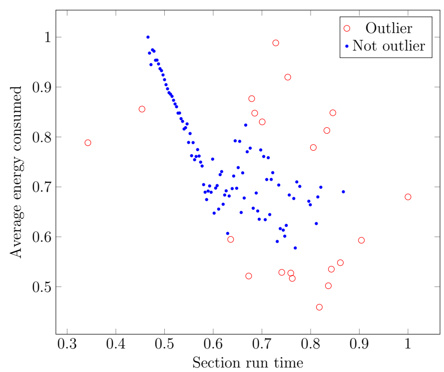


(b) $\varepsilon = 0.15$, $MinPts = 10$

Fig. 9. Comparison of $\varepsilon = 0.15$ on two different data sets



(a) $\varepsilon = 0.08$, $MinPts = 10$



(b) $\varepsilon = 0.1$, $MinPts = 10$

Fig. 10. Comparison of different ε-values on the same data set

4.3 Optimisation Results

The solution proposed in Section 3, will be evaluated based on two examples from a currently working timetable. Both examples were calculated in less than a microsecond using an Intel Core i7-8700K CPU at 3.7GHz with 32GB of RAM. The evaluation will compare the actual energy consumption of the current timetable to the energy consumption in the suggested timetable. The comparison will show how the redistribution of the slack time will achieve a decrease in energy consumption by changing the arrival and departure times of each section. Table 1 shows this comparison. It can be seen that the solution suggests taking all the slack time of the last section from S7 to S8, and 13 seconds from section S6 to S7. This gives a total of 4 minutes and 28 seconds to distribute to other sections. One of the sections receiving more slack time is S4 to S5. By adjusting the slack time from 1 minute and 58 seconds to 3 minutes and 50 seconds, the energy consumption of that section decreased by 33.07%. Another section which benefitted from an increase in slack time is the section from S2 to S3, which resulted in a decrease in energy consumption by 11.71%. Looking at the average total energy consumption of the timetable from S1 to S8, the energy consumption of the current timetable can be decreased by 4.5%. Of course, this decrease is only possible if the suggested timetable is implemented without modification. However, the decision support tool allows the timetable planner to adjust the suggested timetable to fulfil external requirements. For example, it might not be possible for all timetables to remove all the slack time to S8 since this is a central station having trains nonstop arriving or departing. These constraints are not taken into account in this paper. The decision support tool gives the timetable planner the insights to which sections are expensive in energy consumption to take into consideration when modifying a timetable. Table 2 shows a timetable

	Current timetable			**Optimised timetable**			
	Run time hh:mm:ss	Slack hh:mm:ss	Avg. energy consumption	Run time hh:mm:ss	Slack hh:mm:ss	Avg. energy consumption	Difference %
S1-S2	00:11:30	00:01:26	0.753	00:11:57	00:01:58	0.739	-1.86
S2-S3	00:06:00	00:01:36	0.333	00:06:38	00:02:13	0.294	-11.71
S3-S4	00:07:30	00:02:46	0.495	00:08:09	00:03:25	0.464	-6.26
S4-S5	00:05:00	00:01:58	0.251	00:06:52	00:03:50	0.168	-33.07
S5-S6	00:08:00	00:00:31	0.493	00:08:50	00:01:21	0.444	-9.94
S6-S7	00:18:30	00:02:00	0.994	00:18:18	00:01:47	1.000	0.60
S7-S8	00:28:30	00:04:15	0.936	00:24:16	00:00:00	0.952	1.71
Total	01:25:00	00:14:32	4.255	01:25:00	00:14:32	4.061	-4.56

Table 1. Optimised timetable comparison.

containing two sections with no data available. These sections will have an energy consumption of 0 and are constrained to have the same run time as the current timetable. As also seen in Table 1, the solution removes all the slack time from the sections leading to S8. Though, removing all the slack time from a section comes with a high risk. The train cannot be delayed when departing from S8 or on the way towards S9. The train cannot drive any faster, since removing all the slack time sets the run time to be the lowest possible. The suggested timetable shows that for example the section from S16 to S17 and the section from S13 to S14 should be prioritised when distributing slack time to minimise the total energy consumption. The energy consumption of the current timetable can be decreased by 6.23%. The solutions show that the optimised timetables are able to save up to 33.07% energy on a single section and up to 6.23% for a complete timetable.

	Current timetable			**Optimised timetable**			
	Run time hh:mm:ss	Slack hh:mm:ss	Avg. energy consumption	Run time hh:mm:ss	Slack hh:mm:ss	Avg. energy consumption	Difference %
S8-S9	00:07:00	00:01:11	0.630	00:05:49	00:00:00	0.648	2.83
S9-S10	00:07:30	00:00:52	0.917	00:06:38	00:00:00	0.849	-7.42
S10-S11	00:03:00	00:00:48	0.347	00:03:04	00:00:51	0.342	-1.44
S11-S12	00:05:00	00:00:47	0.829	00:04:17	00:00:04	0.880	6.15
S12-S13	00:05:30	00:00:59	0.708	00:05:26	00:00:54	0.712	0.56
S13-S14	00:04:00	00:00:24	0.467	00:05:04	00:01:28	0.358	-23.34
S14-S15	00:03:30	00:00:35	0.605	00:03:49	00:00:54	0.564	-6.78
S15-S16	00:06:30	00:02:47	1.000	00:05:09	00:01:26	0.955	-4.50
S16-S17	00:03:30	00:00:38	0.544	00:04:42	00:01:50	0.376	-30.88
S17-S18	00:03:30	00:00:36	0.595	00:04:06	00:01:11	0.518	-12.94
S18-S19	00:02:30	00:00:18	0.357	00:02:46	00:00:34	0.323	-9.52
S19-S20	00:05:30	00:00:20	0.786	00:05:49	00:00:38	0.755	-3.94
S20-S21	00:03:00	00:00:24	0.450	00:03:26	00:00:50	0.385	-14.44
S21-S22	00:01:30	00:00:00	0.000	00:01:30	00:00:00	0.000	0.00
S22-S23	00:02:00	00:00:00	0.000	00:02:00	00:00:00	0.000	0.00
S23-S24	00:05:00	00:00:56	0.546	00:04:40	00:00:36	0.569	4.21
Total	01:08:30	00:11:35	8.781	01:08:30	00:11:35	8.234	-6.23

Table 2. Optimised timetable comparison.

5 Conclusion and Future Work

This paper has introduced an intuitive decision support tool to support timetable planners in making energy efficient timetables. The tool gives a recommendation based on actual data. The goal was to give timetable planners insight into how train timetables can be energy-optimised based on actual data. To our knowledge, our approach is the first based on actual data from rolling stock.

With this approach, our results show that optimised timetables can save up to 33.07% energy on a single section and up to 6.23% for a complete timetable. Solutions are computed in less than a microsecond.

In future work, we plan to improve the optimisation to produce a more realistic optimum. One improvement is to utilise data from differently configured run times in the timetables to improve the prediction of energy consumption. Another improvement is the constraints for the optimisation. As seen in the results, a solution could choose to remove all slack time for a section. The optimisation model can be given a minimum slack time constraint to solve this. Known constraints such as availability of tracks and platforms could also be taken into account in the optimisation.

Another point for future work is to improve the outlier detection. Alternatives to DBSCAN exist which do not need ε-parameter to be tuned. OPTICS [1] and hierarchical clustering are possible candidates for further investigation. OPTICS build on the same principles of DBSCAN, but it evaluates the local density relative to the individual data points. However, OPTICS needs a maximum ε to reduce computation time. On the other hand, Hierarchical clustering is not density-based, thus, avoids the ε-parameter altogether.

References

1. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data. pp. 49–60. SIGMOD '99, ACM (1999). https://doi.org/10.1145/304182.304187
2. Barrow, K.: Laying the foundations for energy-efficient traction. International Rail Journal (2019), `https://www.railjournal.com/in_depth/energy-efficient-traction/`, [Accessed 2019-06-08]
3. Caprara, A., Fischetti, M., Toth, P.: Modeling and solving the train timetabling problem. Operations Research **50**(5), 851–861 (2002). https://doi.org/10.1287/opre.50.5.851.362
4. Cubris - A Thales Company: (2019), `http://www.cubris.dk/`, [Accessed 2019-06-26]
5. Cucala, A., Fernndez, A., Sicre, C., Domnguez, M.: Fuzzy optimal schedule of high speed train operation to minimize energy consumption with uncertain delays and driver's behavioral response. Engineering Applications of Artificial Intelligence **25**(8), 1548–1557 (2012). https://doi.org/10.1016/j.engappai.2012.02.006
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: 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. pp. 226–231. KDD'96, AAAI Press (1996), `http://dl.acm.org/citation.cfm?id=3001460.3001507`, [Accessed 2019-06-08]
7. Gupta, S.D., Tobin, J.K., Pavel, L.: A two-step linear programming model for energy-efficient timetables in metro railway networks. Transportation Research Part B: Methodological **93**, 57–74 (2016). https://doi.org/10.1016/j.trb.2016.07.003
8. International Union of Railways: Technologies and potential developments for energy efficiency and co2 reductions in rail systems (2016), `https://uic.org/IMG/pdf/_27_technologies_and_potential_developments_for_energy_efficiency_and_co2_reductions_in_rail_systems._uic_in_colaboration.pdf`, [Accessed 2019-06-08]
9. Scheepmaker, G., Goverde, R.M.: 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. pp. 1–20. International Association of Railway Operations Research (2015)
10. Sicre, C., Cucala, P., Fernndez, A., Jimnez, J.A., Ribera, I., Serrano, A.: A method to optimise train energy consumption combining manual energy efficient driving and scheduling. Computers in Railways **114**, 549–560 (2011). https://doi.org/10.2495/CR100511