Master Computer Science

Optimizing the use of
recuperative braking by trams

Name: Rob Mourits

Date: [10/07/2024]

Specialisation: Foundations of Computing

1st supervisor: Walter Kosters
2nd supervisor: Jeannette de Graaf
Ext. supervisor: Marco Hennipman

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Vehicles in a railway system can generate energy by using recuperative braking. This energy can be used by other vehicles under a few restrictions, otherwise the energy is burned and it will be lost. In this thesis, we will attempt to optimize the use of recuperative braking by increasing the dwell time of vehicles at stops. This way they can synchronize with another vehicle, so they can use the recuperative energy. In previous research, several methods have shown the usefulness of optimizing the synchronization time of railway vehicles. We created a greedy heuristic algorithm that can be used in a real-time advisory system that recommends departure times to stationary vehicles. We applied this method to the Avenio tram fleet in The Hague, which showed that approximately 20% of the recoverable burned energy can be saved using this method.

Contents

1 Introduction

Trams are some of the more environmental friendly methods of transport. However, there are still ways to improve the energy efficiency of the tram network. One of the methods that is implemented in most modern catenary (overhead line) based railway systems is the use of recuperative or regenerative braking. A tram can brake using two different mechanisms. The first and most intuitive one is friction or mechanical braking, where the kinetic energy of the tram is transformed into heat and can therefore not be used by the tram anymore. Regenerative braking is a mechanism that reverses the operations of the traction motor [KMB19]. This way the motor functions as a generator and converts the kinetic energy of a tram into electricity while slowing down due to that mechanism. If this energy is then used, for example by another tram that is accelerating at the same time in the same or a close catenary section, this is called *recuperated* energy. Since there is not always another tram in a nearby section that accelerates at the same time as another tram brakes, this can be optimized by synchronizing the driving behaviour of different trams.
In this thesis the usage of recuperated energy by Avenio trams in the tram network of The Hague will be studied. The tram system in The Hague is high traffic, which makes it relatively common for trams to be able to use the energy generated by other trams. However, currently still 9.32% of the recuperated energy generated by these trams is burned. To make the system more efficient the amount of burned energy should be reduced as much as possible.
Several ways to make more efficient use of the recuperated energy will be discussed. An overview can be found in Figure 1. The methods in this figure will be discussed in Section 2.

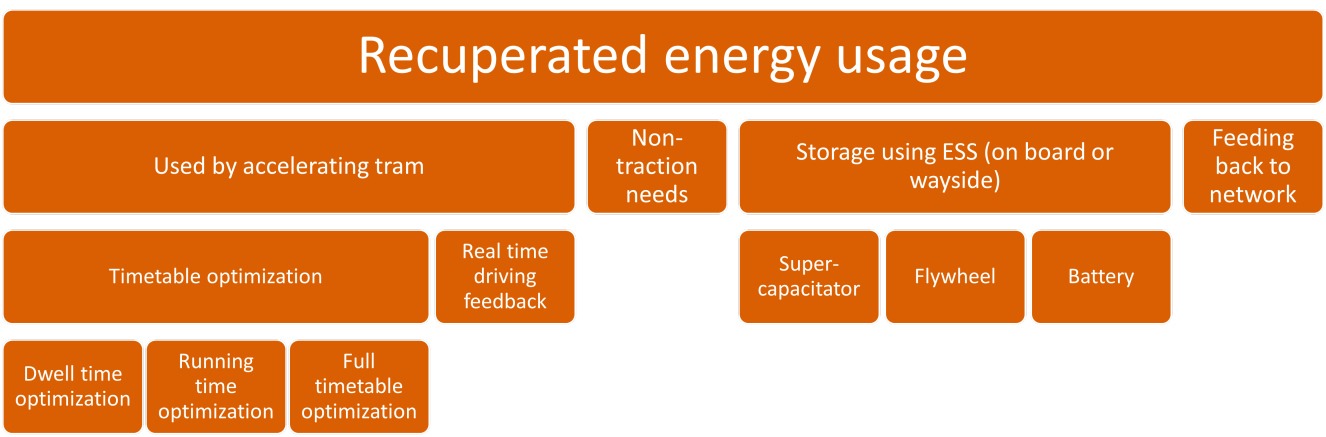

Figure 1: Overview of possible ways to use recuperated energy.

To understand better how recuperative braking works, an example of energy usage of a tram during three minutes can be found in Figure 2. This figure shows that the vehicle will alternate in energy usage while driving from one stop to another. When the vehicle accelerates, the used energy will increase, while the recuperated energy increases when the vehicles brakes. If the recuperated energy increases, this means that there is another vehicle that uses the generated energy for accelerating at the same time. The usage and generation of energy also impacts the voltage on the main line. When energy is used the voltage lowers while it will increase when a vehicle is regenerating energy. When the voltage rises to 720 V, the energy will be burned, since this is the maximum voltage for the main line. In most cases this means that there was no vehicle nearby that used the recuperated energy, which results in the energy being burned. In this case, energy that could have been recuperated will be burned through a resistor on the vehicle.

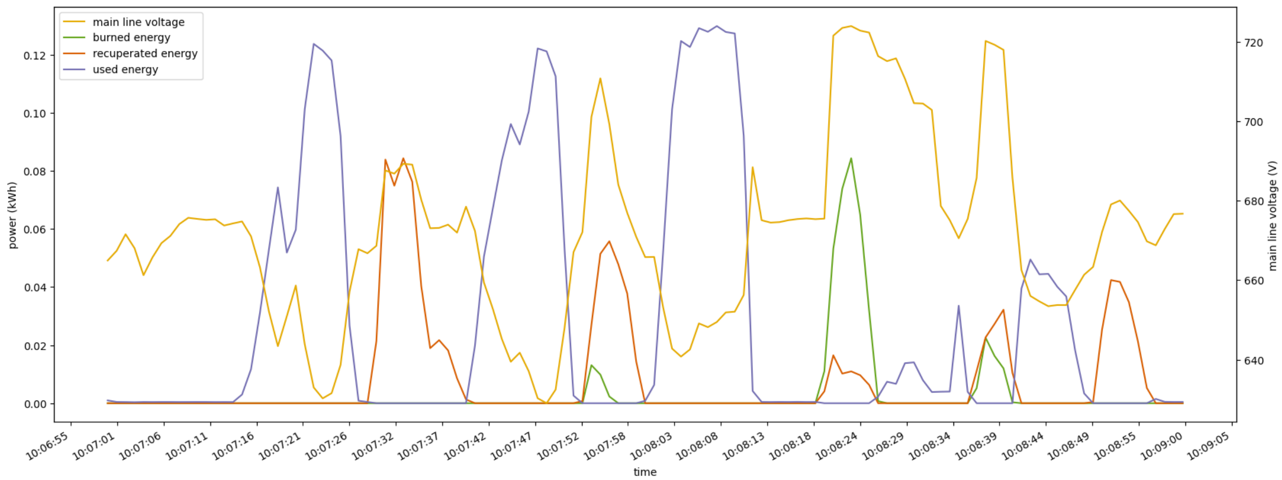

Figure 2: Energy usage of a vehicle and the corresponding main line voltage. On the horizontal axis the time in seconds is displayed, while the vertical axis shows both the energy usage in kWh (left) and the voltage on the main line (right)

In Section 2 we discuss existing research on train and tram energy efficiency. In Section 3 we introduce our method for performing the optimization of the current tram network. In Section 4 we will look at the performance of the algorithm by analyzing performed experiments. In Section 5.1 we will discuss ways to improve and build on this research, and in Section 5 we will draw some conclusions based on the experimental research.
In this thesis we will answer the following research question:

> How can we use a real-time driver advisory system to optimize the use of recuperative braking by trams?

This is a thesis at Leiden University as an internship at Siemens Mobility in Zoetermeer, supervised by Walter Kosters, Jeannette de Graaf and Marco Hennipman.

2 Related work

Two possible methods are usually suggested to perform the optimization of using recuperative energy. The goal of these optimization methods will be to reduce the energy consumption of trams by using the recuperated energy more efficiently.
The first method is by timetable optimization. The timetable that determines when trams arrive and depart from stops can be modified to increase the number of situations where recuperated energy can be used by another tram. This method is used in previous research and shown to be beneficial. Nasri et al. [NMM10] created a genetic algorithm that could save 14% of the energy usage by modifying the stop times. Using mixed integer programming Pena-Alcaraz et al. [PAFC+12] showed that the timetable of the Madrid metro network could be optimized, which can lead to 7% in energy savings. Yang et al. [YNLT14] performed two-objective timetable optimization where they not only showed that it was possible to save 8.86% of the total energy consumption in the Beijing Yizhuang subway line, but also decreased passenger waiting times.
While a modified timetable can save quite some energy, it is important to consider that a timetable is dependent on many different factors, such as passenger load, tram availability and safety. Therefore it is not easy to change a timetable based on just energy savings.
There are also variations on timetable optimization, for example by only influencing the time that the tram waits at a stop, the *dwell time*. This method does not influence the general structure of the timetable and is therefore easier to apply without impacting constraints such as passenger load and tram availability. However, it does come with some constraints, such as the maximum amount of time a modification can differ from the initial timetable. This method has been applied using a greedy heuristic algorithm [FFM14] and has been compared to an evolutionary strategy, using covariance matrix adaptation. This showed that in most cases the heuristic approach was more successful, reducing the total energy consumption by 5.1%. Similarly, [SZJ+23] showed that by modifying the dwell time, the overlap time between an accelerating and braking tram can be optimized. This led to maximum energy savings of 7.39%. A slightly different variation is running time optimization [Alb04], which focuses on increasing or decreasing the speed of the tram for a specific time period to influence the timing of braking and accelerating. Previous research showed that there is a balance between the possible energy savings and the negative impact on the waiting time for passengers [AO02], so with this approach there are still some important timetable constraints to keep in mind.

The second method that can be applied to better use recuperated energy is by giving advice to the drivers of the tram on when to depart or when to brake. This could influence trams by leaving slightly later than planned so that the energy of another tram can be used for traction. A similar system is already studied [DEPV16] for energy efficient driving advice in general, but this system does not consider the use of recuperative energy. For this direct advice approach it is important to understand the way trams are operated. There are four different driving modes [KMB19]. First the tram accelerates, which means it draws energy from the catenary and tries to reach maximum speed. The second and third mode are cruising and coasting. During cruising the driver tries to keep the speed constant by drawing a small amount of energy from the catenary. During coasting almost no energy is used and the train will slowly lose speed. The final mode is braking, which will happen when the tram is close to a stop. In this case the tram delivers energy back to the catenary. In areas with small driving distances between stops, such as The Hague, cruising will rarely be used. Therefore the speed profile of trams consists of accelerating to the maximum speed, coasting for a short time (depending on the distance to the next stop) and braking. This method is the fastest way to get from one station to another. The most energy efficient way would be to coast for as long as possible [SLTG13], since this mode uses hardly any energy from the catenary. Therefore, the most energy efficient method would be to accelerate to a certain speed such that from that speed the vehicle can coast until the next stop is reached. However, since this driving profile is not time efficient there should be a balance between journey time and energy efficiency [BTR10].

In this thesis we focus on optimization with regards to the amount of recuperated energy that is used by another tram. This is done by transferring the recuperated energy back to the catenary. It can then be directly absorbed by another accelerating vehicle that needs energy at that moment for traction. This method is theoretically cost-free, since it already happens in the current tram network. It is only possible for another vehicle to use the recuperated energy if it accelerates in the same or a close energy supply section. If there are no vehicles that can absorb the energy, and no other ways of energy reusage are implemented, the electricity will be burned through resistors in the catenary. This happens if the voltage of the main line is higher than the maximum level, which is controlled by the voltage level control system that is part of the recuperative braking system [PAFC+12].

However, there are also other ways to use the recuperated energy [UKC19b]. First and foremost the vehicle uses energy for needs apart from moving, for example lighting and air conditioning. A small part of the recuperated energy will be used for these non-traction energy needs.

Secondly the energy could be stored if there is an energy storage system (ESS), which stores the additional recuperated energy so it can be used for acceleration later. There are several different ways to store recuperated energy [GGPB13] in an ESS, such as batteries, flywheels and super capacitors. An ESS can also be used as a wayside system, where the energy is stored at a certain location and can be used later by another tram that passes by the wayside ESS. Next to direct storage it is also possible to feed the energy back to the main power network [UKC19a, PB19]. This method however requires the substations to be invertible to feed the power back to the network, which is currently not the case in The Hague.

One of the suggested possible ways to optimize the synchronization between different trams is by giving real-time advice to the driver on when to accelerate or brake. Something similar has been implemented for NS trains to increase the energy efficiency [CJS+23]. This research was not focused on the use of recuperated energy from other vehicles, but rather on energy efficient driving by the vehicle itself and on making sure that the trains arrive on time. While the energy efficiency approach is different, this shows that supplying advice to the driver can be a successful method of influencing the driving behaviour. Also note that for trains the precision of the schedule might be more complex, energy usage per vehicle is lower for trams, and there are likely less vehicles close enough to use recuperated energy. Finally, while trams have to take pedestrians and cars into account, trains are mostly isolated from other traffic.

One factor that influences the energy efficiency of trams is the amount of headway that is used. Headway is the distance of time between two trams on the same track. When the headway is too small it is not safe, since trams are driving too close to each other. A smaller headway however can result in more efficient use of recuperative energy [YNLT14].

Another study was focused on timetable rescheduling in case of a disturbance in the tram network, which leads to delays [YTY+16]. This research considered mainly the additional travel and waiting time for passengers, but also included the energy efficiency of the vehicles. A dynamic programming method was used by approaching the situation as a stochastic decision problem. Similarly, in research on the five largest railway stations in Poland [UKC22], the arrival and departure delays were both considered, together with train energy cooperation.

2.1 Solving methods

The goal of this thesis is to present an optimization method for recuperative energy usage. In literature, there are several popular methods that were applied for this purpose. Every approach consists of a decision variable: the factor that the algorithm tries to influence to improve the energy efficiency. These decision variables can be categorized in timetable optimization, dwell/running time optimization and speed profile optimization [AHPV13]. The latter method does not focus on the use of recuperated energy but on the energy efficiency of the driving pattern in general. However, there are some algorithms that focus on speed profile optimization that can teach us something about optimizing the use of recuperated energy.
Every approach also needs to define what the goal is of the optimization, so an objective function needs to be defined. This can simply be the total energy consumption, which is the most general option, since the end goal is to reduce the amount of energy used. The method is especially useful for speed profile optimization. A bit more specific is recuperated energy usage, which directly focuses on how much of the regenerated energy is used by other vehicles. Even more specific is synchronization time, which tries to optimize the amount of time two trams are in the same section while one is braking and the other is accelerating. This method is easier to implement when the details of the system are not known, but might be a bit too simple, since the goal in the end is to reduce energy consumption, while synchronization time only indirectly influences this.
Next to the energy related objective functions, it is also possible to optimize other variables, such as driver satisfaction or user satisfaction. These functions can depend on the amount of delay [YTY+16] or a comfortable driver profile for example.
The final part of each method is the algorithm that is used to perform the optimization. These algorithms can be categorized in three different algorithm types. The first type is the analytical method [DF12, RPFC08, PAFC+12, LG03]. In many cases this is done using mixed integer optimization. This method reduces the decision variables to a set of integers with a list of constraints describing the situation. An objective function is then defined, and a solver is used to find the set of integers that maximizes the objective function within the search space defined by the constraints. The second method similarly starts by defining a set of constraints and an objective function. This method uses genetic algorithms [YLG+13, BTR10, LL14] to modify the decision variable by performing mutation and crossover. Every solution is then evaluated by the objective function. Both the analytical method and genetic algorithms work with a simplified version of the real situation. This is of course necessary to apply an algorithm, but they do not use knowledge about the situation and natural paths that can lead to a solution. Therefore, [FFM14] proposed a greedy heuristic approach to solve the problem. This method uses the knowledge about the situation to find a good solution within reasonable time. Similar to the genetic algorithm approach, using a greedy algorithm cannot ensure that the optimal solution is found.

3 Method

An analysis on the impact of a real-time advice system is performed. The problem at hand requires very specific modifications to the current situation to save a significant amount of energy. While timetable optimization can cover many different aspects of the problem and could increase the chance of using recuperated energy, it relies heavily on the punctuality of the trams. If a tram arrives or leaves from a stop a few seconds later, this could already have a negative influence on the possibility to use recuperated energy. On average, a sequence in which a vehicle brakes and regenerates energy lasts for only approximately 9 seconds and an acceleration sequence lasts approximately 22 seconds. This shows that a timetable that is accurate to the minute is not good enough to align such small braking and acceleration segments, especially considering deviations from the timetable such as delays. Timetables are also dependent on many other factors, such as passenger load, punctuality and safety. In general, energy efficiency is not included in the set of factors that influence the design of a timetable. Therefore, the only way to realistically influence the driving behaviour of the vehicle with such precision that it can align braking and acceleration segments is to give real-time advice to the driver through a driver-advisory system, such as implemented in [CJS+23].

To find the best possible advice to the driver, a greedy heuristic approach will be used. The biggest advantage of this method is that it draws from the large amount of knowledge about the tram behaviour to find a natural path to a solution. Both an analytical method and the use of genetic algorithms have to generalize the problem on several different aspects, while the problem is very specific. Next to that, a greedy approach can yield faster results than both other methods. The algorithm will be explainable, which is not the case for genetic algorithms, and can therefore easily be customized and improved for slightly different situations. The method will also have higher acceptance due to the explainability.

The objective function will be based on the amount of energy recuperated. Only looking at the synchronization time of trams is a bit too specific for this problem. In the end the goal is to save energy, which makes the amount of energy recuperated directly useful, while the synchronization time only indirectly influences the total energy consumption. Using the total energy consumption as a objective function, however, would be too general for this problem. There are many factors that influence the total energy usage of a tram system, but in this thesis we only focus on the use of recuperated energy. Trying to determine the total amount of energy usage would be out of the scope of this project.

The variable that will be modified to improve the amount of recuperated energy will be the dwell time. As discussed previously, timetable optimization will not be useful in this case. Another possibility would be running time optimization, in which the time that a tram brakes is changed by modifying the driving behaviour. However, it is more difficult for tram drivers to influence the time that they brake since this is dependent on factors such as safety and the fact that they are reaching a stop. It is more realistic to change the dwell time, since the vehicle will be stationary at a stop and can leave roughly whenever is most desirable. It is important to note that there are also limitations when modifying dwell time, since in certain situations it will be essential to depart, for example when another tram is heading to the same stop, or when a tram is already delayed. Similarly, a vehicle might need to stay for longer at a stop if, for example, there are still people boarding or there is another vehicle in front of it. Out of all the possible factors to influence, however, dwell time optimization is the most flexible variable.

3.1 Assumptions and symbols

Before the algorithm will be introduced the necessary assumptions will be explained.
For the driver-advisory system to be implemented, every vehicle must have real-time access to the geographical location of the other vehicles in the tram network. It would also be possible to implement the real-time system without this information, relying on the expected location of vehicles based on the planned schedule, but this will decrease the system's success significantly.
The vehicles must also contain an on-board computer with enough computing strength to compute the real-time advice within a few seconds. If the computation takes more time than the vehicle can spend stationary at a stop, the advice will be too late and it would not be usable. Currently, the Avenio trams are on average stationary for 70 seconds at a stop.
Finally, the performance of the system will be reviewed in the amount of energy saved when every driver follows the given advice. In reality some drivers might ignore the advice.
An overview of the variables used in the algorithm can be found in Table 1. The most important variables are the raw measured data points $M(q,t)$, which contains information about vehicle q at time t. This information is represented as features that can be found in Table 2. Throughout the algorithm, $M(q,t)$ will be augmented by some features that can be found in this table. Note that not every measured data point will contain all these features. For example, a measurement $M(q,t)$ does not contain a value for SFS, but a measurement on a trip does. Similarly, for $M(q,t)$ the variables q and t are used as indices/keys, but for a trip measurement they turn out to be variables, while r, p and s are indices. See Section 3.3 for an explanation on measurements on a trip. Finally, the parameters used in the algorithm can be found in Table 3.

Variable	**Meaning**
$M(q,t)$	measured data point for vehicle q at time t
$P(r,k)$	k-th data point on the planned trip of route r
$T(r,p,s)$	data point for the s-th second on trip p on route r
$EBI(q,k)$	k-th energy burning interval for vehicle q
$PRI(q,k)$	k-th set of possible receiver intervals for vehicle q
SSD	list of substation distances
$S(r)$	geographical starting point of route r
$E(r)$	geographical ending point of route r
Q	set of all vehicle numbers
ES_{tot}	total amount of saved energy
EBR_{tot}	total amount of recoverable burned energy
PES	proportion of energy saved
$DPES$	difference in proportion of energy saved

Table 1: Variables for the algorithm.

3.2 Recuperation distance

It is not always possible for two vehicles to cooperate by letting one vehicle use the recuperated energy of the other. There are a few restrictions the vehicles need to adhere to before they can cooperate.
The energy supply of trams is done through a catenary system. This system is divided into catenary sections that are between 1.1 km and 2.4 km long in the The Hague tram network. Between two

Features	**Meaning**
r	route number
t	time
q	vehicle number
p	trip number
v	vehicle speed
g	geographical coordinate pair (latitude, longitude)
c	catenary section
f	seconds delay (compared to the schedule)
s	seconds from the start of a trip
ϕ	geographical angle (relative to the equator)
ER	energy recuperated
EB	energy burned
EU	energy used
ES	energy saved
SFS	expected number of seconds from start
SFS_d	seconds from start for the best departure time
PTI	planned trip interval
RDT	recommended departure time
DS	delay score

Table 2: Features that are part of a measurement. Notice that some of these features can be added later throughout the algorithm.

adjacent catenary sections a diode or open line interruption is placed to divide the catenary line. From a recuperation perspective both a diode and open line interruption work the same: they make sure that no energy can be transferred from one section to another over the catenary line. However, it is possible for recuperated energy to go from one catenary section to another over a shared substation. Most adjacent catenary sections share such a substation, making it possible for vehicles in two adjacent sections to cooperate. However, there are also components of one or more sections that share substations only within the component, rendering it impossible for them to cooperate with vehicles outside their component.

Another restriction for two vehicles to cooperate is that the main line voltage cannot rise too much. When a vehicle uses regenerative braking, the voltage in the catenary line rises rapidly. If there is an accelerating vehicle in the same section the voltage also lowers immediately, since it is using energy. When no such vehicle is present and the main line voltage reaches 720 V, the regenerated energy will be burned. For the energy to be transferred to another section over a substation, a minimum voltage of 680V needs to be reached. In this case, the voltage in the section of the braking vehicle will lower, while the voltage in the adjacent section will rise. Every time the energy is transferred to another section there is a chance of the voltage reaching 720 V, meaning that the energy gets burned. Next to that, the further the energy is transferred, the more is lost through resistance in the catenary line. Therefore, the further away a possible cooperating vehicle is, the smaller the probability of successful recuperation from one vehicle to another.

Parameter	Meaning
$EB_{\min}$	minimum amount of energy that needs to be burned to be considered
$ER_{\min}$	minimum amount of energy that needs to be recuperated to be considered
$SD_{\max}$	maximum substation distance
$PD_{\max}$	maximum difference between two planned points
$v_{\min}$	minimum speed to be considered a departure
$d_{\max}$	maximum distance between two points to be considered similar
$\phi_{\max}$	maximum difference between two angles to be considered similar
$EB_{\max}$	maximum amount of energy that is burned per second
$REB_{\min}$	minimum amount of required energy that needs to be burned to be considered
$EBS_{\min}$	minimum amount of energy burned to be used as the start of an EBI
EDB	energy delay balance
$f_{\max}$	maximum seconds delay allowed for a vehicle over a full trip
$h_{\max}$	maximum seconds extra delay allowed for a vehicle at a single stop

Table 3: Parameters for the algorithm. The bottom five parameters can be modified to influence the performance, while the top eight are determined values to ensure the algorithms correctness.

3.2.1 Maximum substation distance

Since there are many unpredictable factors influencing the voltage in the main line and the possibility to recuperate, in this thesis we will approximate the chance that two vehicles can cooperate based on the number of substations between their respective catenary sections. To say something about the distance between two catenary sections it is important to know between which sections a connection exists. This can be represented as a network $G(V, E)$ where the vertices V are catenary sections and the edges E represent a connection between two sections through a substation. The network of the catenary sections used by the Avenio trams in The Hague can be found in Figure 3. This network consists of 48 catenary sections, divided in five components: four small ones with three or fewer catenary sections, and one large one which connects the other 39 sections in the network.

To make an estimation on how many substations can be between two cooperating vehicles, we studied the burning behaviour of vehicles with regards to the proximity of possible receivers. It would be useful to study the distance between two vehicles that are actually cooperating, however it is difficult to know for sure which vehicle used the energy recuperated by another vehicle. The burning vehicle simply increases the voltage on the main line and the receiver decreases the voltage without any indication if the energy from the main line originated from a substation or a recuperating vehicle.

However, it is possible to look at a vehicle that burned energy. If a vehicle burns energy, this means that it was not possible to (fully) cooperate with another vehicle. We looked at vehicles that accelerated at the same time as another vehicle burned energy. Theoretically, these vehicles could act as receivers of the energy that was burned. Since the vehicles did not cooperate, this means that the distance between them was too large. We used this distance to make an assumption on the maximum number of substations between two cooperating vehicles.

To do this, first we need to define a burning interval as an ordered list of measurements. A measurement $M(q, t)$ is an aggregation of the data measured between t (inclusive) and $t + 1$ (exclusive) for a single vehicle q. Every measurement $M(q, t)$ contains information about for example the trip number p, location g and energy usage EU. This information will be represented in square brackets, so respectively $M(q, t)[p]$, $M(q, t)[g]$ and $M(q, t)[EU]$. An overview of all features

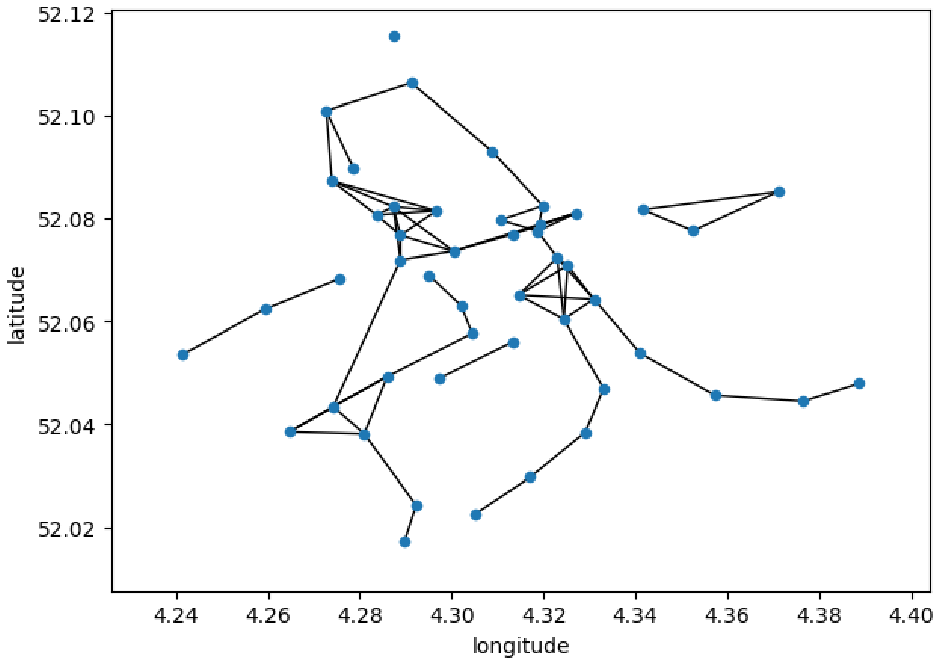

Figure 3: Connections between catenary sections used by Avenio trams in The Hague.

for a measured data point can be found in Table 2. The k-th energy burning interval $EBI(q_0, k)$ is defined as follows:

$$EBI(q_0, k) = (M(q_0, t_k), M(q_0, t_k + 1), \dots, M(q_0, t_k + t'_k)) \tag{1}$$

where

$$\begin{gathered}(EB_{\min} \leq M(q_0, s)[EB] \text{ and} \\ M(q_0, s)[ER] \leq ER_{\min}) \\ \text{or} \\ (EB_{\min} \leq M(q_0, s-1)[EB] \text{ and} \\ EB_{\min} \leq M(q_0, s+1)[EB] \text{ and} \\ M(q_0, s-1)[ER] \leq ER_{\min} \text{ and} \\ M(q_0, s+1)[ER] \leq ER_{\min})\end{gathered} \tag{2}$$

for all $t_k \leq s \leq t_k + t'_k$. In this definition, the interval $[t_k, t_k + t'_k]$ is as long as possible. Since there could be short moments where the amount of energy burned becomes less than $EB_{\min}$ the second part of the or clause is added. This part makes sure that if there is a measured amount of burned energy less than $EB_{\min}$ surrounded by two measurements above $EB_{\min}$ it will still be part of the burning interval. Next to the energy burned, the energy recuperated is also considered. If a vehicle burns energy and also recuperates, it means that there is actually a receiver, however, the amount of energy burned is more than the receiver can use. Since we only want to study situations where there is a possible receiver but it is too far away, we only consider the situations where energy is burned and not recuperated. To illustrate when something is considered an energy burning interval EBI, Figure 4 shows the burned and recuperated energy over time and how the energy burning interval is determined.

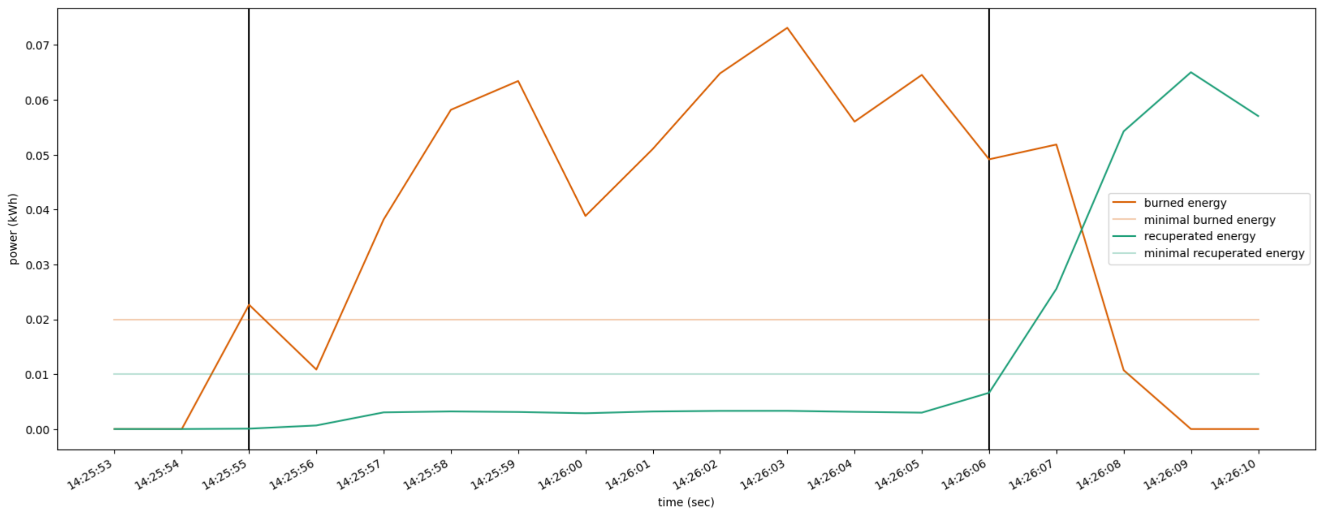

Figure 4: A burning interval bounded by the amount of energy burned and recuperated, as defined in Equation 1 and 2. On the horizontal axis the time in seconds is displayed, while the vertical axis shows the amount of energy burned and recuperated in kWh.

In this figure we see that the interval starts at 14:25:55 and ends at 14:26:06. Since we allow a single second of lower energy, 14:25:56 is included in the interval, even though it is lower than the minimal amount of burned energy. At 14:26:07 the amount of recuperated energy rises above the maximum and since it stays there for at least another second this is not included in the interval, even though the amount of burned energy was still higher than the threshold.

For every vehicle it will be checked if it could have been a possible receiver of the burned energy at that time. If this is the case, the interval of energy usage will be defined as an element of $PRI(q,k)$, the set of possible receiver intervals for burning interval $EBI(q,k)$. An interval of measurements is only considered to be an element of $PRI(q,k)$ when more energy is used by the receiver than burned by the supplier at the start, the end and in total over the entire interval. A typical element $(M(q_1,t), M(q_1,t+1), \ldots, M(q_1,t+t'))$ of $PRI(q_0,k)$ satisfies:

$$\begin{aligned}
&(M(q_0,t)[EB] \leq M(q_1,t)[EU] \text{ or} \\
&\ M(q_0,t)[EB] \leq M(q_1,t+1)[EU]) \\
&\text{and} \\
&(M(q_0,t+t')[EB] \leq M(q_1,t+t')[EU] \text{ or} \\
&\ M(q_0,t+t')[EB] \leq M(q_1,t+t'+1)[EU]) \\
&\text{and} \\
&\sum_{i=t}^{t+t'+1} M(q_0,i)[EB] \leq \sum_{i=t}^{t+t'+1} M(q_1,i)[EU]
\end{aligned} \tag{3}$$

with $EBI(q_0,k) = (M(q_0,t), M(q_0,t+1), \ldots, M(q_0,t+t'))$. Since there could be a short delay in energy transfer, burning and accelerating can happen at the same timestamp or with a single second delay. The first and second clause of the *and*-statement ensure that there is more usage than burning at the start and at the end of the burning interval, respectively. The final clause ensures that in total more energy is used than burned, also with a possible second delay.

The list of substation distances SSD between the catenary sections of all burning vehicles and their

possible receivers can be constructed by computing $substation\_distance(M(q_0, t)[c], M(q_1, t)[c])$ for every possible receiver interval by vehicle q_1 in all instances of $PRI(q_0, k)$.
In this definition, $substation\_distance(c_0, c_1)$ returns the number of substations that need to be crossed from catenary section c_0 to c_1, which is equal to the number of edges between two nodes in the catenary section connectivity network. If c_0 and c_1 are not connected at all, $substation\_distance(c_0, c_1) = \infty$.
Now SSD contains all the substation distances between a burning vehicle and a vehicle that could have been a receiver. The frequencies of these distances can already give some insight into some distances occurring more than others. However, some distances will occur more than others due to the nature of the graph. For example, if there are two longer catenary sections the distance between these sections will have a higher frequency than other distances. Therefore this figure must be compared to the distance between two random vehicles at a random time. Figure 5 shows the proportion of both the distances between a burning and possible receiving vehicle and between two random vehicles in the Avenio fleet in The Hague. Situations where the substation distance is infinite (since the sections are not connected) are excluded, since those vehicles will never be able to cooperate

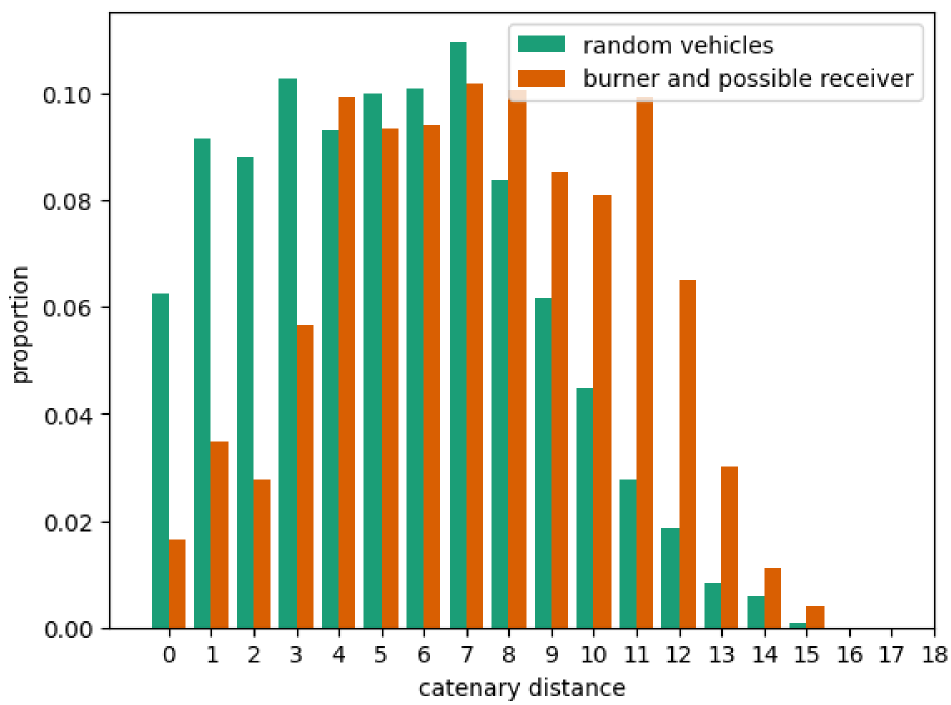

Figure 5: Proportion of the number of substations between a burning vehicle and a possible receiver and the proportion of the number of substations between two random vehicles.

It is visible that for smaller substation distances the proportion of random vehicles is larger than for burner and receivers. From a distance of four substations the balance shifts the other way. This shows that for distances less than four, there is a smaller chance that there will be a burning vehicle and a possible receiver. The reason for this is most likely that most braking vehicles will not burn energy when there is a possible receiver less than four substation away, since the braking vehicle can cooperate with the receiver and recuperate energy. This suggests that on average, the maximum distance $SD_{\max}$ between two catenary sections with cooperating vehicles is three substations.

3.3 Average route

To determine the recommended dwell time, an estimation must be made about when vehicles will use and recuperate energy. Next to that it is important to find the time it takes for a vehicle to move from a certain point to another in order to properly estimate the location of the vehicle in the future based on the current location. It would be possible to make this estimation based on the timetable and look at the planned time difference between locations and the positions before stations to estimate where a vehicle will most likely burn energy. However, it will be more similar to the real situation if these estimations are based on measured data. Therefore, to make this estimation it is most accurate to look at previous situations where this trip was performed by a vehicle. In our case study this is done based on a single week of measured data. If this system is applied in a real situation, it might be best to compute the average route based on recent data.

From this data, trips are defined for each route. A trip by a specific vehicle q is defined as an ordered list of measurements $T(r,p,s)$. Every instance of $T(r,p,s)$ corresponds to exactly one instance of $M(q,t)$. The main difference between two corresponding values of $T(r,p,s)$ and $M(q,t)$ is the indexing. For $T(r,p,s)$ the tram number and the exact time are not relevant. Therefore indexing is done relevant to the specific trip, using s as the index of the measurement, which equals the number of seconds from the start of the route r. The route is a line number, which means that multiple trams can drive on the same route r. Furthermore, p is the trip number, which is a unique number for each time a trip on route r is driven.

Since the measured GPS coordinates are not always accurate, all measured point on the trips will be mapped to the closest point on the route as planned. To do this, we need to define a planned route $P(r,k)$, which consists of a path that passes through an ordered list of GPS coordinate pairs and is constructed from a separate dataset. In $P(r,k)$, k is the index of the point on route r. Every instance of $P(r,k)$ contains the geographical coordinate pair of this point, $P(r,k)[g]$ and the angle of the vehicle on this point $P(r,k)[\phi]$.

The distance between two points $P(r,k)$ and $P(r,k+1)$ should not be too large, since this will decrease the accuracy of the data. Therefore a maximum point distance $PD_{\max}$ is defined. If the distance between two points is larger than $PD_{\max}$, additional points will be added using interpolation.

Initially, $P(r,k)$ only contains the coordinate pair and angle of the point, but throughout the algorithm, more features will be added. For example, for each value of $P(r,k)$, the expected number of seconds from start $\overline{SFS}$ and the average amount of energy burned $\overline{EB}$ at that location will be determined.

The start and end of trips are determined based on the location and angle of the vehicle (the direction the vehicle is facing). A trip starts when the vehicle has started moving. Only full trips from the start to the end of a route are considered. Since measured data sometimes contains faulty or missing measurements, determining the trip is done with some margin. A specific measurement $M(q,t)$ will be considered the geographical trip starting point $S(r)$ of route r if the following constraint is satisfied:

$$dist(M(q,t)[g], P(r,0)[g]) \leq d_{\max} \text{ and } |M(q,t)[\phi] - P(r,0)[\phi]| \leq \phi_{\max} \tag{4}$$

where $dist(x,y)$ is defined as the Euclidean distance between the geographical location of points x and y, $d_{\max}$ is the predefined maximum distance the measured point can have from the actual starting point and $\phi_{\max}$ is the maximum difference between the angle of the measured point and the angle of the actual starting point for the point to still be considered as a starting point of the trip. However, the geographical starting point will not directly be used as a trip starting point, since a vehicle can wait at the first stop for a long time, which will increase the time that a vehicle is driving on a trip if the geographical starting point is used. Therefore the trip only starts when the vehicle reaches a certain minimum speed $v_{\min}$ after leaving the geographical start. Therefore the start of a trip $T(r, p, 0)$ is equal to $M(q, t_{\min})$, where $t_{\min}$ is chronologically the first timestamp after t_0 with $M(q, t_0) = S(r)$ where $v_{\min} < M(q, t_{\min})[v]$.

The first time the end of the route is reached is determined similarly to Equation 4, but using $P(r, m_r)$ instead of $P(r, 0)$ where m_r is the final point on the planned trip of route r, and $E(r)$, which is the geographical end of route r. The measurement at the end of the trip will be denoted as $M(q, t_{\max})$.

Trip number p at route r from the first measurement where $k = 0$ until $k = t_{\max} - t_{\min}$ can now be defined as an ordered list of measurements:

$$(T(r, p, 0), T(r, p, 1), \ldots, T(r, p, t_{\max} - t_{\min})) \tag{5}$$

is in one-to-one correspondence with

$$(M(q, t_{\min}), M(q, t_{\min} + 1), \ldots, M(q, t_{\max})) \tag{6}$$

Now that the start and end point of a trip are defined, every measurement on a trip can be connected to its closest point on the planned trip. Every point on the planned trip will contain a set of measured points that are closest to that planned trip point. This set will be defined as $P(r, k)[C]$, the set of closest measurements on trips for $P(r, k)$:

$$\begin{aligned} P(r, k)[C] = \{T(r, p, s) \mid dist(T(r, p, s)[g], P(r, k)[g]) \leq dist(T(r, p, s)[g], P(r, k')[g]) \\ \text{for all } k' \neq k)\} \end{aligned} \tag{7}$$

With the set of closest measured points for each planned point, the expected number of seconds from the start of the trip $\overline{SFS}$ and the average amount of energy burned $\overline{EB}$ at a point can be computed:

$$P(r, k)[\overline{SFS}] = \frac{1}{|P(r, k)[C]|} \left(\sum_{T(r,p,s) \in P(r,k)[C]} s \right) \tag{8}$$

$$P(r, k)[\overline{EB}] = \frac{1}{|P(r, k)[C]|} \left(\sum_{T(r,p,s) \in P(r,k)[C]} T(r, p, s)[EB] \right) \tag{9}$$

For every GPS coordinate pair on the planned route the average is taken over the number of seconds a vehicle at this location has been driving on its trip. Similarly, the average amount of energy burned at this location is computed.

3.4 Recommended departure time

Now that we have the necessary data structures we can start computing the recommended departure time, which can be used as advice for drivers when stationary at stops.

In the explanation of the algorithm, we will refer to the braking vehicle as the *supplier*, since it supplies regenerative energy that can be used by another vehicle. This other vehicle will be called the *receiver*, since it receives the energy supplied by the supplier and uses it while accelerating.

The driver-advisory systems works as follows. When the speed of a vehicle reaches 0, it will be considered a (possible) receiver and the recommended departure time will be computed. The

advisory system gets as input the measured data $M(q_{\rm rec}, t_0)$ of the receiving vehicle $q_{\rm rec}$ at t_0, the moment it becomes stationary. The output of the driver-advisory system will be the recommended departure time $M(q_{\rm rec}, t_0)[RDT]$.
There are two different recommendations the system can give. The first one is to depart as soon as possible, as happens currently in the tram network. In this case $M(q_{\rm rec}, t_0)[RDT]$ will be 0. If this is not the case, the advice will be to increase the departure time slightly by staying at the stop for a specific number of seconds longer if there is a possible supplier. In this case $M(q_{\rm rec}, t_0)[RDT]$ will be the recommended departure timestamp.
The algorithm approaches the situation from the receiver's perspective, since by modifying the dwell time, the moment of used energy (accelerating from a stop) is influenced. The first goal is to find the most suitable supplier for a receiver, if present. Determining the recommended departure time consists of three parts. First the simple constraints for the receiver and a possible supplier will be determined. After that some details about the supplier's current and future position and energy usage will be added. Finally, based on these details, the best supplier is selected and the recommended departure time is computed.

3.4.1 Simple constraints

Not for every receiver it is possible to increase the dwell time. There are two very simple constraints for the receiving vehicle. The first is that it is currently stationary at a stop, which should automatically be handled, since the algorithm will only be used if this is the case. The second constraint is that the current number of seconds the vehicle is behind on schedule must be smaller than the maximum delay $f_{\max}$ on a trip. The maximum delay for the trip is a parameter that should be determined beforehand. The second constraint is represented in the following definition:

$$M(q_{\rm rec}, t_0)[RDT] = 0 \text{ if } f_{\max} \leq M(q_{\rm rec}, t_0)[f] \tag{10}$$

During the different steps of the algorithm a set of possible suppliers $M(q_{\rm rec}, t_0)[PS]$ is used. An element of PS is a single measurement $M(q_{\rm sup}, t_0)$ of a possible supplier $q_{\rm sup}$ at the same time t_0 as the receiver. For a vehicle to be a possible supplier the only simple constraint is that the catenary sections of the supplier and receiver should be connected over less than or equal to $SD_{\max}$ substations:

$$M(q_{\rm rec}, t_0)[PS] = \{M(q_{\rm sup}, t_0) \mid \mathit{substation\_distance}(M(q_{\rm rec}, t_0)[c], M(q_{\rm sup}, t_0)[c]) \leq SD_{\max}\} \tag{11}$$

Now $M(q_{\rm rec}, t_0)[PS]$ only contains the measurements of vehicles that are close enough to possibly cooperate.

3.4.2 Supplier details

To determine if a possible supplier is a good match for the receiver, it is important to know the current situation of this vehicle. Under the assumption that the GPS coordinates and route number of these vehicles are available it is possible to determine where on the route these vehicles are. Using the planned route constructed in Section 3.3, the expected time on their route can be determined. First we find the closest point $P(r_0, k)$ on route r_0 for every measurement $M(q_{\rm sup}, t_0)$. This closest point $M(q_{\rm sup}, t_0)[p_{\rm close}]$ satisfies:

$$\mathit{dist}(M(q_{\rm sup}, t_0)[g], P(r_0, k)[g]) \leq \mathit{dist}(M(q_{\rm sup}, t_0)[g], P(r_0, k')[g]) \text{ for all } k \neq k' \tag{12}$$

Using this closest point we can find the value of SFS.

$$M(q_{\text{sup}}, t_0)[SFS] = p_0[SFS] + \\ (dist(p_0[g], M(q_{\text{sup}}, t_0)[g])/dist(p_0[g], p_1[g])) * (p_1[SFS] - p_0[SFS]) \tag{13}$$

where

$$\begin{aligned} (p_0 = M(q_{\text{sup}}, t_0)[p_{\text{close}}] \text{ or } p_1 = M(q_{\text{sup}}, t_0)[p_{\text{close}}]) \text{ and} \\ M(q_{\text{sup}}, t_0)[g] \text{ is geographically between } p_0[g] \text{ and } p_1[g] \text{ and} \\ p_0 = P(r_0, k) \text{ and } p_1 = P(r_0, k+1) \text{ and} \\ M(q_{\text{sup}}, t_0) \in M(q_{\text{rec}}, t_0)[PS] \text{ and } r_0 = M(q_{\text{sup}}, t_0)[r] \end{aligned} \tag{14}$$

In this definition the location of vehicle q_{sup} is found with regards to its planned route. The location of the vehicle will approximately be between two planned points p_0 and p_1. The expected seconds from start SFS is then calculated using the SFS values of the two points on the planned trip and the distance between the vehicle's location and those two points.
The next step is to estimate the vehicle's behaviour for the coming $h_{\max}$ seconds. This interval of the planned trip $M(q_{\text{sup}}, t_0)[PTI]$ will be defined as $(P(r, k), P(r, k+1), \ldots, P(r, k+k'))$, which satisfies:

$$\begin{aligned} k = \min\{x \mid M(q_{\text{sup}}, t_0)[SFS] \leq P(r, x)[SFS]\} \text{ and} \\ k + k' = \max\{y \mid P(r, y)[SFS] \leq M(q_{\text{sup}}, t_0)[SFS] + h_{\max}\} \text{ and} \\ M(q_{\text{sup}}, t_0) \in M(q_{\text{rec}}, t_0)[PS] \text{ and } r = M(q_{\text{sup}}, t_0)[r] \end{aligned} \tag{15}$$

This statement defines the interval of the trip from the current timestamp to $h_{\max}$ seconds later for vehicle q_{sup}. The first point on the interval is simply chosen as the first point on the trip of route r with a value of SFS that is higher than $M(q_{\text{sup}}, t_0)[SFS]$. The end of the trip is defined in the same way but for the last point before $M(q_{\text{sup}}, t_0)[SFS] + h_{\max}$. If the current delay $M(q_{\text{sup}}, t_0)[f]$ is already smaller than $f_{\max} - h_{\max}$, instead of $h_{\max}$, $f_{\max} - M(q_{\text{sup}}, t_0)[f]$ will be used in Equation 15, to avoid creating a higher delay than $f_{\max}$.

3.4.3 Best supplier

Every point on the planned trip interval will be evaluated as a possible moment where q_{sup} can cooperate with q_{rec}. To do this, a delay score DS is defined:

$$P(r, k)[DS] = \\ EDB * (P(r, k)[\overline{EB}]/EB_{\max}) + (1 - EDB) * (1 - (P(r, k)[\overline{SFS}] - M(q_{\text{sup}}, t_0)[SFS])/h_{\max}) \\ \text{if } P(r, k)[\overline{EB}] > REB_{\min} \text{ for } P(r, k) \in M(q_{\text{sup}}, t_0)[PTI] \tag{16}$$

This definition creates a balance between the amount of energy that could be saved and the seconds of extra delay this will cause. Here EDB is a variable that controls this balance. If it is set to 1, this means that the algorithm will always choose the option with the largest energy gain, in this definition represented as the expected amount of energy burned divided by the maximum amount of energy that can be burned at a single point. If EDB is set to 0 it will only try to minimize the maximum delay, which is computed by taking 1 minus the extra delay divided by the maximum allowed delay for a single stop. For every second we also define at what time the receiving vehicle should depart to make sure that this second (of possible burning) can be covered by the energy usage of the receiver. This time is defined as SFS_d, the seconds from start of the supplier's route that is ideal for departure:

$$P(r,k)[SFS_d] = \min\{P(r,k')[SFS] \mid EBS_{\min} \leq P(r,k'')[\overline{EB}] \text{ for } k' \leq k'' \leq k\} \tag{17}$$

This definition finds the earliest possible timestamp from which the amount of energy burned will constantly be at least $EBS_{\min}$. Using both the delay score of a burning peak and the departure time to actually use this burning peak, the best departure time can be chosen:

$$\begin{aligned} M(q_{\text{rec}},t_0)[RDT] &= t_0 + \min\{(P(r,k)[SFS_d] - M(q_{\text{rec}},t_0)[SFS]), 0\} \\ \text{where } P(r,k)[DS] &= \max\{P(r,x)[DS] \mid P(r,x)[DS] \in M(q_{\text{sup}},t_0)[PTI] \\ &\qquad \text{for all } M(q_{\text{sup}},t_0) \in M(q_{\text{rec}},t_0)[PS]\} \end{aligned} \tag{18}$$

This definition selects the recommended departure time for q_{rec} out of all the different suppliers and all their timestamps. Note that the decision to delay is not communicated between trams. This means that two trams might both attempt to use the energy of a single supplier, which increases the chance of energy being saved.

3.5 Simulation

To evaluate the effect that the real-time system has on the energy usage, a simulation is used, which is based on measured data for a limited time interval, for example a single day. Using this we can simulate the behaviour of a set of vehicles if the real-time feedback system was used. This is done by iterating over all the moments that one of the vehicles is stationary. For each of these moments the recommended departure time RDT is computed according to Section 3.4. If RDT is not 0, the delay will be simulated by shifting this vehicle's data such that the previous departure time is changed to RDT:

$$M(q,t) = M(q,t-\Delta t) \text{ if } M(q,t_0)[RDT] = t_0 + \Delta t \text{ and } t_0 \leq t - \Delta t \tag{19}$$

$$M(q,t) = M(q,t_0) \text{ if } M(q,t_0)[RDT] = t_0 + \Delta t \text{ with } t - \Delta t \leq t_0 \leq t \tag{20}$$

In this equation every value of $M(q,t)$ is shifted if there is a recommended delay at t_0 where $t_0 < t - \Delta t$. Here t_0 is the time that a delay of Δt seconds is inserted. The gap that is created between t_0 and RDT will be filled with copied features from $M(q,t_0)$, as shown in Equation 20. The only feature that will not be simply copied is the delay f, since it should increase with the new delay:

$$M(q,t)[f] = M(q,t)[f] + \min(\Delta t, t - t_0) \text{ if } M(q,t_0)[RDT] = t_0 + \Delta t \text{ and } t_0 \leq t \tag{21}$$

When a vehicle is stationary for longer than $2 * f_{\max}$, which will likely happen between two trips, the value of $M(q,t)[f]$ is reset to the value of f in the original $M(q,t)$. At this point the vehicle can catch up on the extra delays before the next trip starts. This means that the shifts that are performed before are also reversed, meaning that the vehicle will be starting the next trip at exactly the same time as in the original $M(q,t)$ without the added delays. After the shifts in the data are performed and the delays are updated, the simulation will continue to the chronologically next moment a vehicle is stationary to compute the delay. This continues until the end of the simulation.

3.6 Performance evaluation

To evaluate the energy savings of the day, after adding the delays, it would be ideal to directly look at the amount of energy that can be saved. However, due to the complex nature of the electricity network, this is not possible with the available data. As mentioned in Section 3.2, we cannot determine if two vehicles can cooperate purely based on the data. Therefore we must create an estimation of how much energy could be saved as a result of the delays.
For every second and every vehicle we will compute the theoretical energy saved in both the original data and the modified data with added delays:

$$\begin{aligned} M(q_0,t)[ES] = \min(M(q_0,t)[EB], M(q_1,t)[EU], |ER_{\min} - M(q_0,t)[ER]|) \text{ where} \\ \mathit{substation\_distance}(M(q_0,t)[c], M(q_1,t)[c]) \leq SD_{\max} \text{ and} \\ M(q_1,t)[EU] > M(p,t)[EU] \text{ for all } p \neq q_1 \end{aligned} \tag{22}$$

Using the energy savings per second per vehicle we can compute the total amount of theoretical energy savings by summing over all vehicles in a time interval between a fixed t_{start} and t_{end}:

$$ES_{\text{tot}} = \sum_{q \in Q} \sum_{t=t_{start}}^{t_{end}} M(q,t)[ES] \tag{23}$$

Next we will compute the total amount of recoverable energy by summing the values of EB where ER is under the $ER_{\min}$ limit. At the start of Section 4 more will be explained on the definition of recoverable energy.

$$EBR_{\text{tot}} = \sum_{q \in Q} \sum_{t=t_{start}}^{t_{end}} M(q,t)[EB] \;\; \text{if } M(q,t)[ER] \leq ER_{\min} \tag{24}$$

Now we can compute the proportion of energy that is saved:

$$PES = ES_{\text{tot}}/EBR_{\text{tot}} \tag{25}$$

As mentioned, there can be situations where theoretically energy could have been saved, but in reality the energy was still burned. This means that the value of ES_{tot} of the original unaltered data will not be 0, which is what could be expected. To include this in our estimated savings we should subtract PES_o, the original PES, from the value of PES after delays, PES_d. Therefore the objective function, which should be maximized, is defined as the difference in proportion of energy saved:

$$DPES = PES_o - PES_d \tag{26}$$

4 Results

To evaluate the performance of the system several experiments were conducted. These experiments were performed on measured data of the Avenio tram fleet in The Hague in the week from November 13 to 19, 2023. In this week the Avenio fleet consisted of 56 active vehicles. These vehicles drive on 6 out of 13 lines in the entire The Hague tram network, namely line 2, 9, 11, 15, 16 and 17. These routes pass 121 stops in total. Every vehicle is equipped with sensors that measure 532 different signals. The data measured is aggregated per second per vehicle and can therefore directly be used as data points $M(q,t)$, for vehicle q at timestamp t, see Section 3.

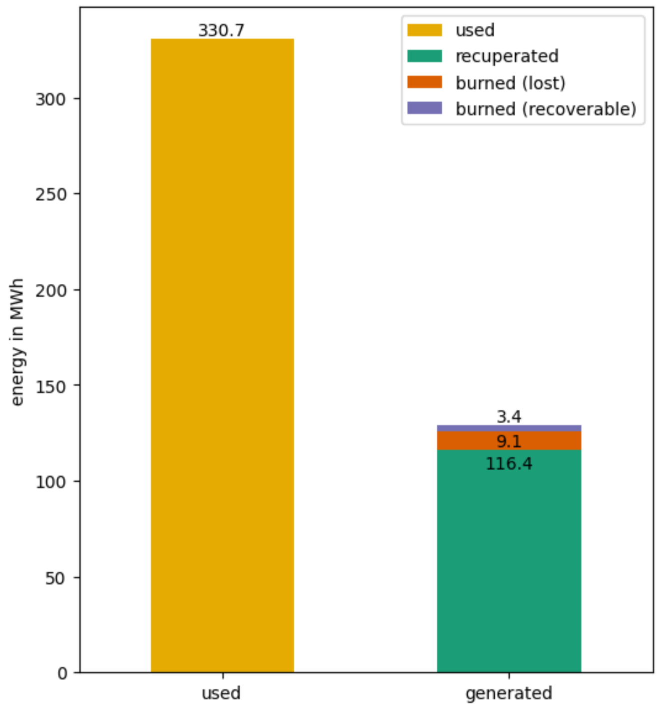

Figure 6: Energy usage and generation by the Avenio fleet during a single week.

The total energy usage of the Avenio fleet in the mentioned week can be found in Figure 6. This figure shows how much energy currently is used, generated and burned in a week, without using the real time system to optimize recuperated energy usage. In this figure the used energy is the total amount of energy that is used by the vehicles for traction purposes. The recuperated energy is the energy generated by regenerative braking that was successfully used by another vehicle. The two burned categories represent all the regenerated energy that was not used by other vehicles and therefore had to be burned.

However, we can define a stricter upper bound for the maximum amount of energy that can be saved than simply using the total amount of burned energy. When a vehicle burns energy but at the same time also regenerates this shows that there is a receiving vehicle. Since some of the energy is still burned, this shows that likely the voltage on the main line is already quite high, meaning that it is not possible to recuperate, even if there is a receiving vehicle. Therefore, some of the generated energy is recuperated, but the rest is burned. In these cases there is no point in trying to find a possible receiver using the algorithm, since there already is a valid receiver, so the energy is lost. Therefore the only part of the burned energy that could theoretically be saved is the energy that is burned without energy being recuperated at the same time, the recoverable burned energy. For the Avenio fleet this means that at most 3.4 MWh could theoretically be saved in a week, which is equal to 2.95% of all the generated energy.

4.1 Parameter settings

The default parameter settings used in the experiments can be found in Table 4. Some of the parameters can be modified to influence the performance of the algorithm, while the value of other parameters should be based on the situation and not be modified. We will explain why we assigned the given values as shown in Table 4 to the parameters that should not be modified.

Parameter	**Value**
$EB_{\min}$	0.02
$ER_{\min}$	0.01
$SD_{\max}$	3
$PD_{\max}$	10
$v_{\min}$	30
$d_{\max}$	0.0005
$\phi_{\max}$	20
$EB_{\max}$	0.055
$REB_{\min}$	0.03
$EBS_{\min}$	0.002
EDB	0.9
$f_{\max}$	105
$h_{\max}$	60

Table 4: Parameters for the algorithm.

First of all, $EB_{\min}$ should be set to a value that ensures that significant burning is taking place. Similarly, $ER_{\min}$ should be a value that is high enough to make sure that insignificant measurements are not considered, but low enough to include every time that there is a significant amount of recuperated energy. The value of SD_{max} is already discussed in Section 3.2. For creating the planned route $PD_{\max}$ needs to be defined. This value is chosen to be high enough to make sure that in most situations it takes less than a second for a vehicle to travel from one point to the next while driving away from a stop. The values of $v_{\min}$, $d_{\max}$, $\phi_{\max}$ are determined by manually recognizing routes driven by Avenio vehicles based on the measured data. The values are chosen such that these routes are all included with the correct starting time. The value of $EB_{\max}$ is determined by simply finding the maximum value of all points $P(r,k)$ for every route r used by the Avenio fleet and every index k.
The remaining parameters, $REB_{\min}$, $EBS_{\min}$, EDB, $f_{\max}$ and $h_{\max}$ can be modified to study the performance of the algorithm. In all experiments where the values of these parameters are not specifically mentioned they are set to their default values from Figure 4.

4.2 Days of the week

The driving schedules for trams can differ per day of the week and time of the day. In the weekend for example, there might be fewer trams than during the week. Next to that some days might be busier than others, and external causes such as the weather and traffic jams could affect the amount of recuperation. In these experiments we study a single week of tram activity, from Monday, November 13, 2023 to Sunday, November 19, 2023. Figure 7 shows the total amount of recoverable energy that was burned in this week, together with the amount of energy that could have been saved if the real-time driver advisory system was used. The amount of saved energy is computed by multiplying $DPES$ (as computed in Section 3.6) by the total amount of recoverable energy burned.

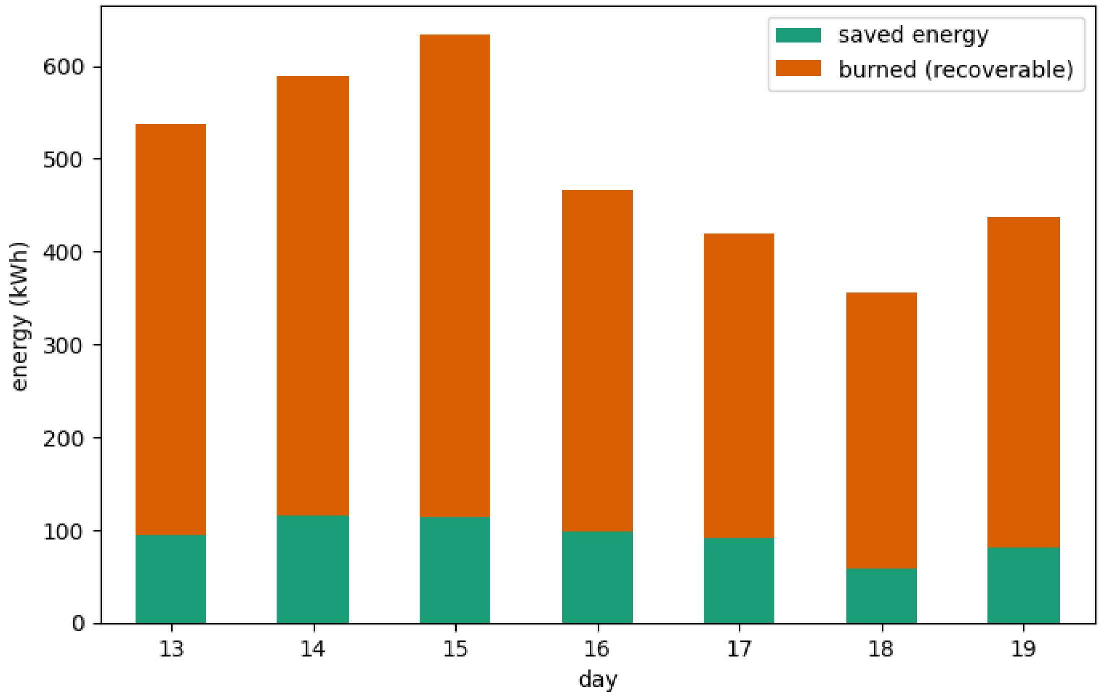

Figure 7: Total amount of recoverable energy burned per day and the amount of energy that could have been saved using the real-time advisory system.

First of all, we can see the large difference in absolute energy burned over the different days. On Wednesday the 15th more than 600 kWh is burned, while on Saturday the 18th this amount is approximately 350 kWh. The value of $DPES$, however, which represents the proportion of burned energy that could have been saved, is similar for most days. On average the value of $DPES$ is 0.20, which is 20% of all recoverable burned energy. This means that in this week 650.94 kWh can be saved when using the real-time advisory system, and still 2784.64 kWh is burned. To know why this energy was not saved we studied the reasons for energy loss and other details of the inserted delays for a single day.

4.3 Delay details

The reasons of energy loss on November 13, 2023 after inserting delays using the algorithm can be found in Figure 8. The reasons are split into seven different categories. Note that the percentages in the figure are with regards to the total amount of recoverable burned energy, as explained in Figure 6.

saved The most important category is *saved*. This represents all the energy that would not have been burned on this day if the real-time system was used. Since we only look at the energy

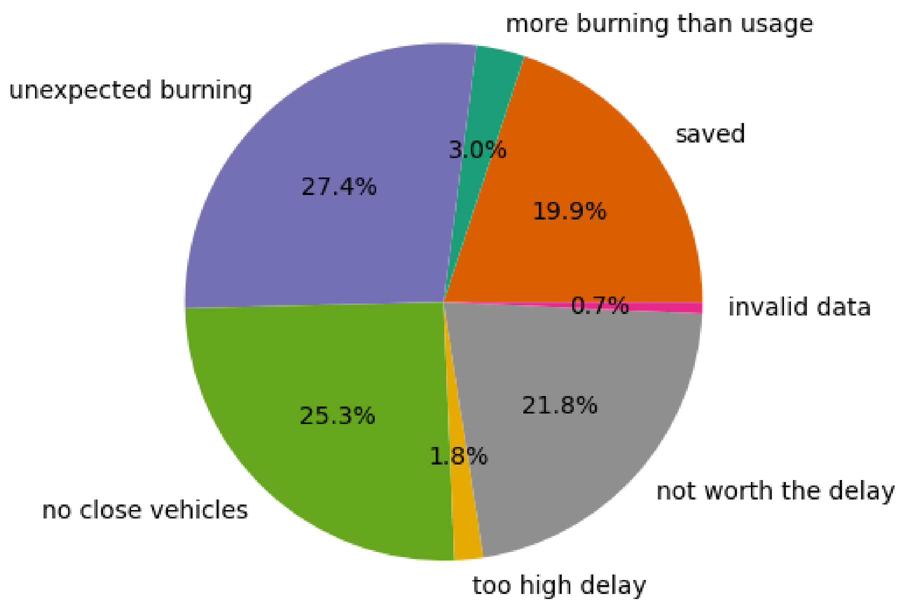

Figure 8: The reasons that recoverable burned energy could not be saved after using the real-time driver advisory system.

burned in the delayed dataframe, we cannot compare to the original situation in which no extra delays were added. Therefore, the percentage shown in *saved* is PES_d and not $DPES$, which means it is a slightly larger percentage than the real amount of energy that can be saved.

unexpected burning Then there are some categories where burning is nearly unavoidable due to the nature of the situation. The largest category is *unexpected burning*, which means that energy was burned at a location where on average no large amount of energy is burned. This can be due to the driving style of the tram driver or unexpected situations such as people crossing the tram tracks, but not at a pedestrian crossing.

no close vehicles Roughly a quarter of the time a vehicle burns energy when expected, but there are *no close vehicles* within the range of $SD_{\max}$ catenary sections at that time, or at most $h_{\max}$ seconds earlier. This means that there was no possible vehicle that could have cooperated with the burning vehicle.

more burning than usage One of the fewer occurring reasons for energy loss is *more burning than usage*, which means that there was some cooperation between vehicles, but the amount of energy burned by the supplier was larger than the amount of energy used by the receiver. Therefore, some of the energy is still burned.

invalid data The smallest category is *invalid data*, which means that the burning vehicle had some technical problems and could therefore not correctly send their location to possible receivers, rendering it impossible for the receivers to modify their delay time to cooperate with this vehicle.

too high delay The burning in the final two categories might be possible to reduce by modifying the parameters of the algorithm. One is the *too high delay* category, which represents the cases in which there are possible receivers close enough, but they are already delayed more than $f_{\max}$, meaning that they cannot delay more. This category will always exist since some vehicles are already delayed too much without even using the system. However, the system also increases the delay, since it gives the advice to vehicles to depart later. The size of this category is quite small, since 105 seconds every trip seems to be enough to insert most useful delays. It might be possible to decrease the size of this category even more by choosing better moments to delay.

not worth delay Decreasing the size of the previous category could, however, increase the final category, *not worth the delay*. This is simply all the energy that is burned that does not fit in any of the other categories. This likely means that it was theoretically possible to save the energy burned, but the system decided not too recommend delays to possible receivers since either the amount of energy that could be saved was too low or the delay itself was too long. In these cases the system decides that it is not worth the delay to save that small amount of energy.

To have a better overview of the changes made in the schedule we look at the length of the inserted delays. The distribution of the length of the inserted delays can be found in Figure 9.

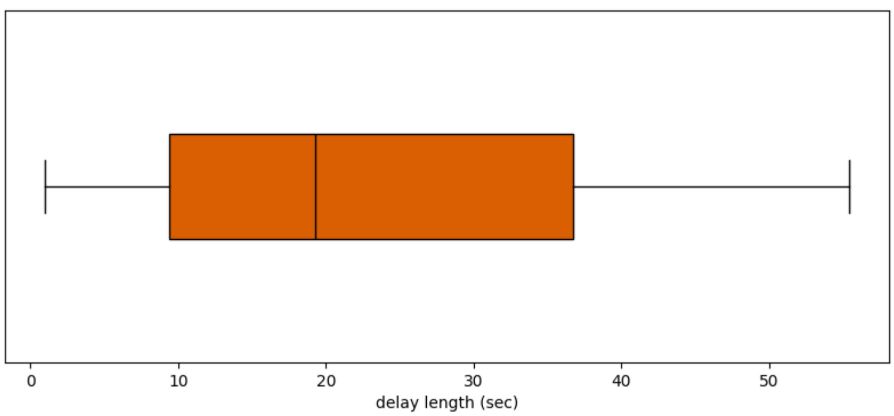

Figure 9: The length of delays in seconds.

This figure shows that most delays are between 10 and 40 seconds long, with slightly more shorter delays, with the median being 20. The reason for this is that shorter delays have a larger chance of being inserted due to the fact that *EDB* is not equal to 1 and because some longer delays might not be possible if the delay exceeds $f_{\max}$. The longest delays are a little bit less than 60 seconds since this is the value used for $h_{\max}$ and therefore the upper limit. In general the delays are evenly divided over the day, since every trip the total delay is reset.
In the following experiments we will study the effect of some parameter settings for November 13.

4.4 Energy delay balance

One of the most influential parameters is the energy delay balance, *EDB*, introduced in Section 3.4.3. When *EDB* gets closer to 1, the algorithm will choose to delay longer if more energy can be saved, while a lower EDB will only perform delays when a significant amount of energy can be saved or when the delay is small. When EDB is too low the algorithm will (almost) never insert delays. Therefore we study the effect of an EDB of 0.3 or higher. The results of this experiment can be found in Figure 10.

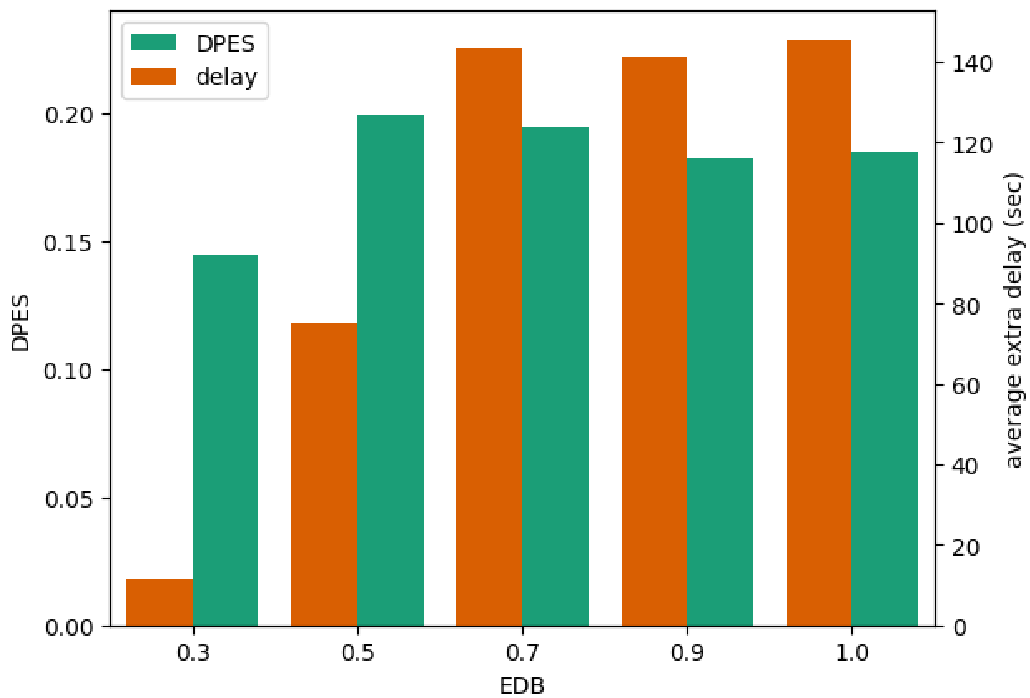

Figure 10: The average delay in seconds per vehicle and the difference in proportion of energy saved ($DPES$) for different values of EDB.

As expected, when the value of EDB is higher, both the amount of energy saved and the average delay per vehicle will increase. However, the maximum amount of energy that can be saved seems to be reached when EDB is set to 0.5. This is likely due to the fact that some delays are inserted that do not save much energy but create quite some extra delay. This means that later during that day some other delays that could save more energy might not be possible anymore since the total delay f on that day is already too high.
Between an EDB of 0.5 and 0.7 we see a significant rise in average extra delay, while $DPES$ decreases slightly. This shows that a higher EDB does not guarantee more energy savings. An EDB of 0.5 might be the best choice, since it has the highest amount of energy savings while also creating relatively little extra delays.

4.5 Minimum burning

Another parameter that influences when a delay is inserted is $REB_{\min}$, the required amount of energy that needs to be burned at a location to be considered significant enough burning to delay another vehicle. If this value is too high very few delays will be inserted, but if it is too low, delays will be recommended for an insignificant amount of energy burned, which could also result in less energy savings, since the maximum allowed delay $f_{\max}$ will be wasted on small burning amounts. The performance for different values for $REB_{\min}$ can be found in Figure 11.

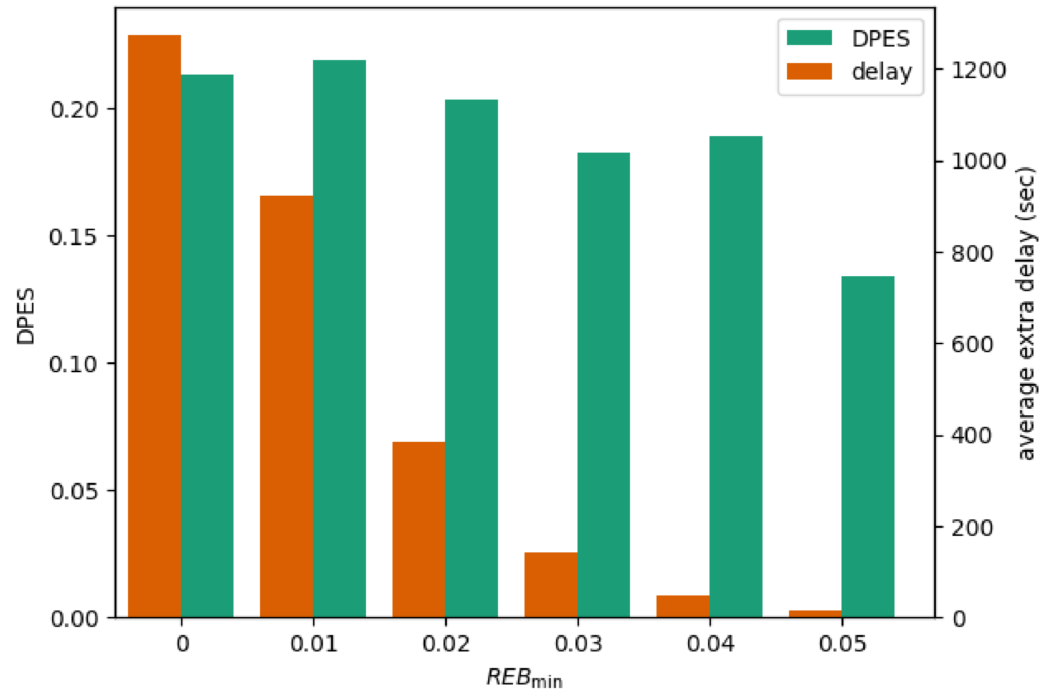

Figure 11: The average delay in seconds per vehicle and the difference in proportion of energy saved ($DPES$) for different values of $REB_{\min}$.

The $DPES$ scores with the different values for $REB_{\min}$ are relatively similar, all between 0.18 and 0.22, with $REB_{\min} = 0.05$ as the only outlier. This shows that if $REB_{\min}$ is too high, some burning peaks will be excluded while they could have been avoided by inserting delays. For values lower than 0.04 the performance is relatively similar. The reason for this is that EDB also has influence on which burning peaks are high enough to delay for. However, there is a slight decrease in $DPES$ if $REB_{\min}$ is 0. This shows that using a burning threshold that is not too low does have a positive effect on the amount of energy saved.

Next to the performance, the average delay per vehicle is shown. This is the total seconds of delay added divided by the number of active vehicles. We see that for a low $REB_{\min}$ the delay is very high, since many attempts to save energy are performed by adding new delays. Higher values of $REB_{\min}$ result in less delay time since only high burning peaks are avoided. Therefore it might be better to take a slightly higher value of $REB_{\min}$, such as 0.02, since there is much more delay with only a slight increase in saved energy.

After the algorithm decides to insert a delay to cover a burning peak, the ideal departure time is determined, as explained in Equation 17. This is done by finding the moment from which the energy will be constantly higher than $EBS_{\min}$ (minimum amount of energy burned at the start of a burning interval), until the selected burning peak. Essentially, $EBS_{\min}$ is used to find the start of the burning interval containing the burning peak. We looked at the performance and delay of the algorithm with different values for $EBS_{\min}$. The results can be found in Figure 12.

While there is some difference in performance, the value of $EBS_{\min}$ does not appear to have a large impact. In most cases the burning peak will be covered by the usage of the receiving vehicle, meaning that no or little energy needs to be burned. When the value of $EBS_{\min}$ is higher the burning interval will be shorter, meaning that the receiver will have to delay a bit longer since the

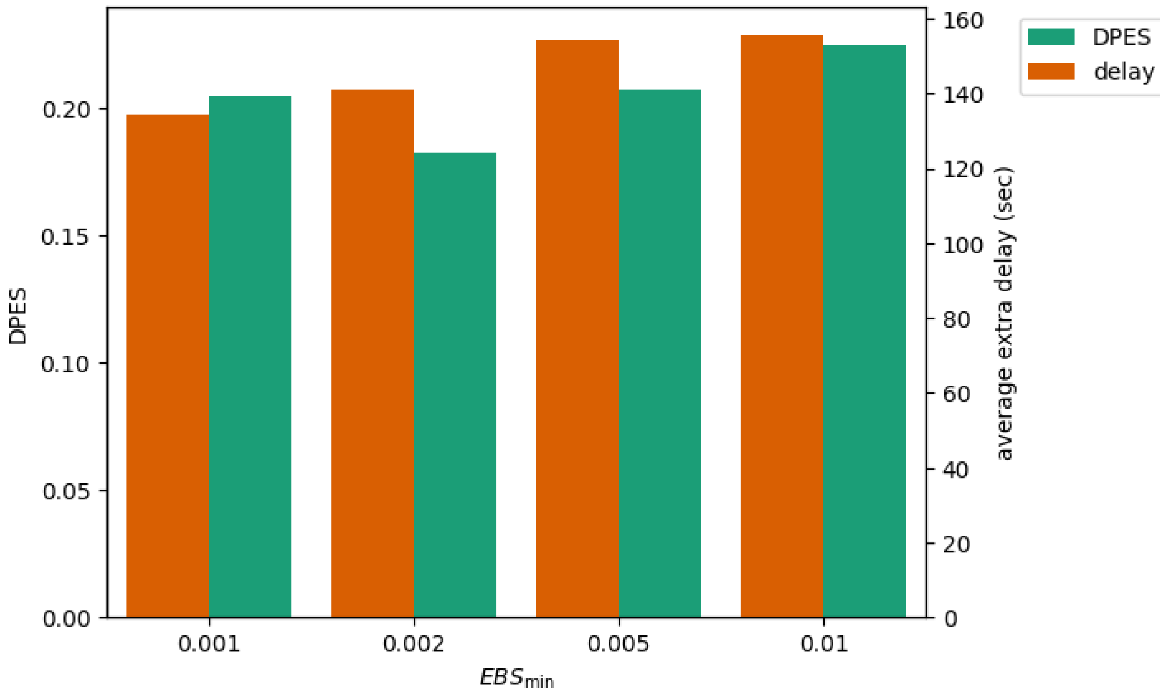

Figure 12: The average delay in seconds per vehicle and the difference in proportion of energy saved $(DPES)$ for different values of $EBS_{\min}$.

departure time will be later. Therefore the delay is slightly higher for higher values of $EBS_{\min}$, but not much.

4.6 Maximum delay time

The parameter settings discussed can all influence the performance of the trams within the current rules and infrastructure of the tram network in The Hague. However, it is also interesting to study the effect of the real-time advisory system in a situation with slightly different rules. One thing we studied is the effect of the maximum waiting time at a single stop $h_{\max}$ and the maximum total delay time on a trip $f_{\max}$.

The maximum allowed total delay on a trip is 105 seconds, since this value is used by the Avenio trams as a limit to determine if a tram is delayed or not. The maximum extra delay at a single stop is a bit less strict, but cannot be set too high since this might reduce traveler comfort. Therefore it is set to 60 seconds. Now we will look at the influence these limits have on the performance if they are set to higher values, shown in Figure 13.

One thing that immediately stands out is that higher values for $f_{\max}$ and $h_{\max}$ do not necessarily mean that the performance will be better. A reason for this can be that $h_{\max}$ also directly influences the value of DS. As can be seen in Equation 16, the second part of the sum is the extra delay divided by $h_{\max}$, so if this value is higher this means that longer delays can have higher scores. If longer delays are inserted and $f_{\max}$ stays the same, this might decrease the number of delays that can be inserted, and therefore possibly decrease the amount of energy that can be saved.

More surprising is the fact that a higher $f_{\max}$ value with the same $h_{\max}$ values can decrease the performance. Theoretically, this should rarely be the case, since the only place where $f_{\max}$ is used is

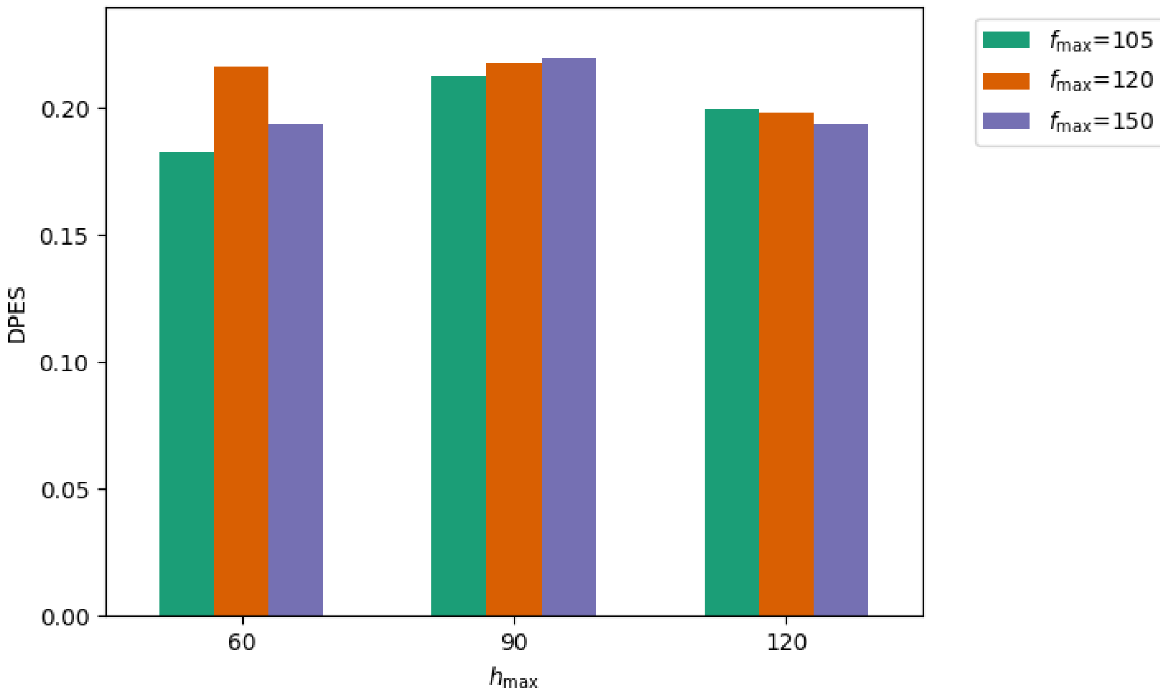

Figure 13: The difference in proportion of energy saved ($DPES$) for different values of $h_{\max}$ and $f_{\max}$.

as a selection criterion when choosing if a delay can be inserted. However, as with every parameter that influences the delays, it can happen that a certain delay can be chosen that turns out not to be a good choice. For example, due to the high $f_{\max}$, maybe a very long delay is chosen, but the peak is not covered due to unexpected burning. In this case it was better to not take that delay, which would have happened with a lower $f_{\max}$, so that smaller delays could have been inserted later.
Another reason that higher values of $h_{\max}$ and $f_{\max}$ do not always increase the performance is that only 1.8% of the burned energy is not saved due to the delay being too high, as shown in Figure 8. This means that we can change the upper limits of the amount of seconds delay, but there is not too much energy that can be gained by doing that.

4.7 Different catenary section network

One of the reasons that energy is burned is because there are no close vehicles to cooperate with. This occurs frequently due to the way the catenary sections are connected, as seen in Figure 3. There are four components that are not connected to the main component, which means that if there is burning in those components there is a smaller chance of possible cooperation: 43.3% of all the energy that is burned is burned in one of those nine unconnected components. Therefore we will study the performance if every adjacent catenary section is connected by a shared substation (meaning that a vehicle can move from one to the other without any section in between). This results in the catenary section network shown in Figure 14.
In this version of the network 13 additional connections are added between sections that originally were not connected by substations. If this hypothetical catenary section network would have been used much would change with regards to energy usage. It will influence when energy gets recuperated

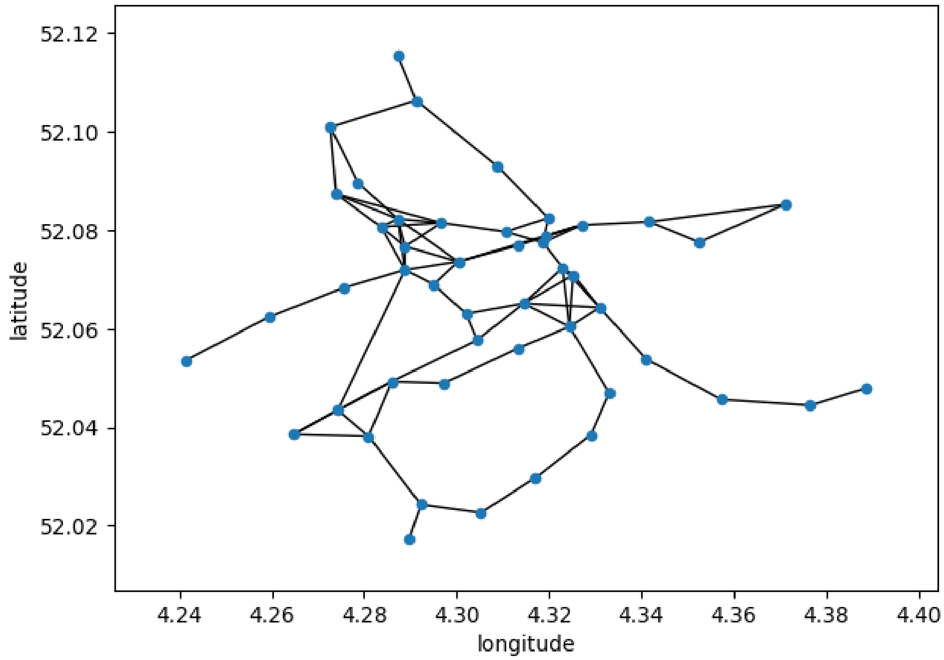

Figure 14: Connections between catenary sections used by Avenio trams in The Hague if every pair of adjacent sections shares a substation.

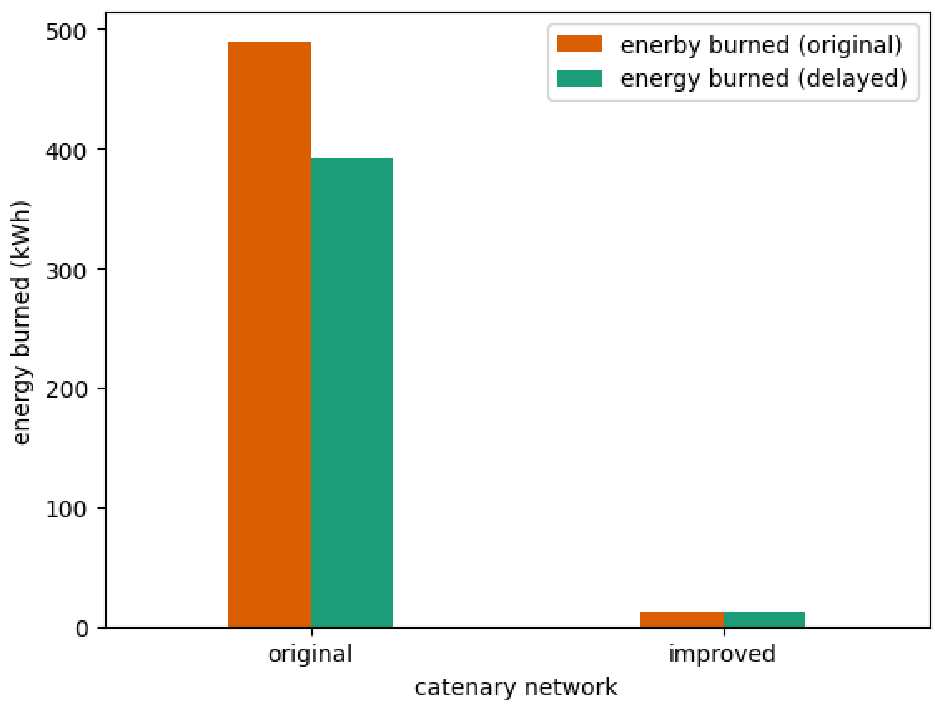

Figure 15: The expected amount of burning before and after adding delays for both the original (Figure 3) and the improved catenary network (Figure 14).

or burned and therefore also the voltage on the main line. Therefore we could never fully simulate this situation using the data and tools used for creating the real-time feedback system. However, we can still theorize about the energy usage in this situation based on our cost function computation. Looking at $DPES$ would be misleading since the computation of PES_o is calculated under the assumption that the original data uses the original catenary section network. Therefore we will look only at the absolute values of the recoverable energy that is still burned, with and without the real-time advisory system. Note that these values are an approximation under the assumption that all recoverable burning is avoided if a vehicle within $SD_{\max}$ substations is using energy at the same time. As explained in Section 3.2, this is not always the case. The results of the simulation can be found in Figure 15.

It is clear from this figure that a fully connected catenary network would likely avoid almost all burning, but using the real-time advisory system will not have much influence on the improved system, since it is already quite optimized. The reason for this large difference is that almost every time energy is burned the vehicle is in one of the smaller four components of the network. If these components are connected there will almost always be a vehicle within $SD_{\max}$ substations distance that uses energy while another vehicle generates energy. However, it might not be possible to modify the current network due to infrastructural limitations.

5 Conclusion

In this thesis we studied the energy efficiency of trams, specifically focusing on the use of recuperated energy. To increase the amount of energy that is saved using recuperative braking, a real-time driver advisory system is created. This system uses the data measured in the past to estimate the location of large energy burning peaks that occur when a vehicle is braking, but there is no other vehicle close by to use this energy. Using this information, delays can be inserted. These delays make a vehicle that is waiting at a stop depart a little later than planned, such that the energy they use for accelerating is supplied by another vehicle that would have burned the regenerated energy in the original situation.

The algorithm used in this real-time advisory system uses a heuristic greedy approach. Every time that a vehicle is stationary at a stop the algorithm is used to determine the best choice at that moment. We tested the algorithm using a case study concerning the Avenio fleet in The Hague. The algorithm could save approximately 20% of the recoverable burned energy in this tram network. Over a week this is equal to roughly 651 kWh, which is not a very large amount. This is due to the fact that the Avenio fleet does not burn much energy in general due to the large density of trams in The Hague, which makes the chance of recuperation significantly larger. Next to that, a large part of the burned energy is already not recoverable, due to unexpected burning, no close vehicles, or situations in which the vehicle is already recuperating, but burning at the same time, showing that the voltage on the main line is already too high to recuperate.

We also looked at some scenarios where the situation is slightly different. It turned out that increasing the maximum waiting time or maximum total delay does not necessarily mean that the performance will be better. In the hypothetical situation that all adjacent catenary sections are connected by a substation, however, the amount of saved energy likely increases significantly.

The research question was: *How can we use a real-time driver advisory system to optimize the use of recuperative braking by trams?* To answer this question we created such a system that supplies a recommended departure time based on estimated cooperation between trams. This system, if used adequately, can be used to decrease the amount of energy that is burned.

In conclusion, in a system where more energy is burned than in the Avenio fleet, a real-time driver advisory system for using recuperated energy better could be a good choice to use some of the energy for traction of another vehicle. However, in the Avenio fleet, this might not be the best choice, since very little energy can be saved due to the already efficient tram network.

5.1 Further research

One of the drawbacks of this research is the limitations of the case study. The Avenio fleet is already relatively energy efficient, making it hard and less useful to reduce the amount of burned energy. Next to that, to create a more complete overview, other trams in the The Hague tram system should be included, since they can also cooperate with the Avenio trams.

To give a more accurate estimation of the impact of the real-time advisory system, a complete simulation of the tram network would be a useful tool. In this research it was chosen not to use such a simulation tool since not all the necessary data was available. This could however give significantly more accurate results, since such a tool can include the effect of delays and recuperation on the main line voltage in the simulation. Such a tool can also help to give a more accurate estimation of the amount of burned energy in a fully connected network, as explained in Section 4.7.

Another way to improve the accuracy of the advisory system is to find a better way to determine the start and end of a trip. In this research this is estimated using the location, angle and speed of a vehicle (as shown in Section 3.3), but there will always be outliers that are excluded or included as a valid trip, but should not have been. If the start of a trip would be accurately tracked, this could help create more accurate time and burning estimations.

It could also be useful to make the algorithm less greedy by looking further in the future and estimating the best combination of delays. This would create a much more complex problem that might not result in more energy savings due to the fact that such an algorithm will rely on estimations much further in the future. However, if successful, it could prove to be a more energy efficient solution.

It might also be useful to study better ways to estimate the burning peaks of the vehicles instead of just taking the average route. Since a large amount of energy is lost through unexpected burning, this could improve the algorithm.

Next to the technical improvements, it is also necessary to study the effect of inserting delays on both the drivers and the passengers. If the delays decrease comfort significantly, it might not be a good choice to implement this system.

References

[AHPV13] A.R. Albrecht, P.G. Howlett, Peter J. Pudney, and X. Vu. Energy-efficient train control: From local convexity to global optimization and uniqueness. *Automatica*, 49(10):3072–3078, 2013.

[Alb04] T. Albrecht. Reducing power peaks and energy consumption in rail transit systems by simultaneous train running time control. *Advances in Transport*, 15:885–894, 2004.

[AO02] T. Albrecht and S. Oettich. A new integrated approach to dynamic schedule synchronization and energy-saving train control. *Computers in Railways VIII*, pages 847–856, 2002.

[BTR10] Y.V. Bocharnikov, A.M. Tobias, and C. Roberts. Reduction of train and net energy consumption using genetic algorithms for trajectory optimisation. In *IET Conference on Railway Traction Systems (RTS 2010)*, pages 1–5, 2010.

[CJS+23] A. Cunillera, H.H. Jonker, G.M. Scheepmaker, W.H.T.J. Bogers, and R.M.P. Goverde. Coasting advice based on the analytical solutions of the train motion model. *Journal of Rail Transport Planning & Management*, 28(100412), 2023.

[DEPV16] S. La Delfa, S. Enjalbert, P. Polet, and F. Vanderhaegen. Eco-driving command for tram-driver system. *The 13th IFAC Symposium on Analysis, Design, and Evaluation of Human-Machine Systems HMS 2016*, 49(19):444–449, 2016.

[DF12] F. Fages D. Fournier, D. Mulard. Energy optimization of metro timetables: A hybrid approach. *The 18th International Conference on Principles and Practice of Constraint Programming*, pages 7–12, 2012.

[FFM14] D. Fournier, F. Fages, and D. Mulard. A greedy heuristic for optimizing metro regenerative energy usage. In *Proceedings of the Second International Conference on Railway Technology: Research, Development and Maintenance*, number 240, 2014.

[GGPB13] A. González-Gil, R. Palacin, and P. Batty. Sustainable urban rail systems: Strategies and technologies for optimal management of regenerative braking energy. *Energy Conversion and Management*, 75:374–388, 2013.

[KMB19] M. Khodaparastan, A.A. Mohamed, and W. Brandauer. Recuperation of regenerative braking energy in electric rail transit systems. *IEEE Transactions on Intelligent Transportation Systems*, 20(8):2831–2847, 2019.

[LG03] R. Liu and I.M. Golovitcher. Energy-efficient operation of rail vehicles. *Transportation Research Part A: Policy and Practice*, 37(10):917–932, 2003.

[LL14] X. Li and H.K. Lo. An energy-efficient scheduling and speed control approach for metro rail operations. *Transportation Research Part B: Methodological*, 64:73–89, 2014.

[NMM10] A. Nasri, M. Fekri Moghadam, and H. Mokhtari. Timetable optimization for maximum usage of regenerative energy of braking in electrical railway systems. In *International Symposium on Power Electronics, Electrical Drives, Automation and Motion, SPEEDAM 2010*, pages 1218–1221, 2010.

[PAFC+12] M. Pena-Alcaraz, A. Fernández, A. Cucala, A. Ramos, and R. Pecharromán. Optimal underground timetable design based on power flow for maximizing the use of regenerative-braking energy. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 226:397–408, 2012.

[PB19] M. Popescu and A. Bitoleanu. A review of the energy efficiency improvement in DC railway systems. *Energies*, 12(6):1092, 2019.

[RPFC08] A. Ramos, M. Pena, A. Fernández, and A. Cucala. Mathematical programming approach to underground timetabling problem for maximizing time synchronization. *Dirección y Organización*, 35:88–95, 2008.

[SLTG13] S. Su, X. Li, T. Tang, and Z. Gao. A subway train timetable optimization approach based on energy-efficient operation strategy. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):883–893, 2013.

[SZJ+23] P. Sun, C. Zhang, B. Jin, Q. Wang, and H. Geng. Timetable optimization for maximization of regenerative braking energy utilization in traction network of urban rail transit. *Computers & Industrial Engineering*, 183(109448), 2023.

[UKC19a] M. Urbaniak and E. Kardas-Cinal. Optimization of energetic train cooperation. *Symmetry*, 11:1175, 2019.

[UKC19b] M. Urbaniak and E. Kardas-Cinal. Optimization of using recuperative braking energy on a double-track railway line. *Transportation Research Procedia*, 40:1208–1215, 2019.

[UKC22] M. Urbaniak and E. Kardas-Cinal. Optimization of train energy cooperation using scheduled service time reserve. *Energies*, 15(1):119, 2022.

[YLG+13] X. Yang, X. Li, Z. Gao, H. Wang, and T. Tang. A cooperative scheduling model for timetable optimization in subway systems. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):438–447, 2013.

[YNLT14] X. Yang, B. Ning, X. Li, and T. Tang. A two-objective timetable optimization model in subway systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):1913–1921, 2014.

[YTY+16] J. Yin, T. Tang, L. Yang, Z. Gao, and B. Ran. Energy-efficient metro train rescheduling with uncertain time-variant passenger demands: An approximate dynamic programming approach. *Transportation Research Part B: Methodological*, 91:178–210, 2016.