Demand-responsive Scheduling in Railway Transportation

Christoph Grüne[1][a] and Stephan Zieger[2][b]
[1]Department of Computer Science, RWTH Aachen University, Aachen, Germany
[2]Institute of Transport Science, RWTH Aachen University, Aachen, Germany

Keywords: Integer Programming, Railway Scheduling, Demand-responsive Transport, Rural Railway Networks, Dial-a-Ride, Timetabling.

Abstract: Rural rail transportation can contribute significantly to achieving climate goals and encountering mobility challenges, especially by reactivating currently disused railway lines. Rural areas are mainly characterised by their dispersed demands. Thus, small highly automated rail vehicles could be operated on-demand and thus service-oriented. The operation of those networks is complex and the economic efficiency must be correspondingly high. Therefore, optimised resource planning is necessary. The paper focuses on the planning of a-priori known transport requests.
The paper presents a formulation for the underlying Integer Programming mathematical model that optimises travel times and number of vehicles used under consideration of railway specific constraints such as headway times and deadlock prevention. The modelling goes beyond existing Dial-a-Ride approaches and adds the necessary routing constraints for rail systems as well as energy management constraints for potential refuelling or recharging.
The potential for application of the approach is evaluated in a computational study. A validation scenario shows in an exemplary manner on the one hand how the constraints affect routing on a single track railway line and on the other hand how solving the model with a black-box solver such as Gurobi is handled for this scenario. On a real-world railway line, it can be shown that the Integer Programming solver is able to induce meaningful results for limited input sizes. Further potential improvements are discussed as well.

1 INTRODUCTION

The Commission of the European Union (European Commission, 2020) identified several mega trends that will influence general life in the coming decades. These include challenges such as climate change, urbanisation, the ageing population as well as transport and behavioural changes. In order to mitigate or even avoid the consequences, the strengthening of the railway system is an essential factor. Especially in rural areas, transport services are often poor, non-existent or discontinued. As demand in rural areas is usually much lower and more dispersed, smaller automated rail vehicles are a viable option such as the ANT system proposed by Siemens (Schlaht et al., 2018) or the Aachen Rail Shuttle (Schindler and von Stillfried, 2020).

These small automated trains can operate on a demand-responsive basis, thereby improving both accessibility and availability of public transport. In this paper, technical and legal challenges are assumed to be solved and the focus is on operational questions. The emphasis here is on the scheduling, routing and energy management of the vehicles. In contrast to road transport, rail transport has some other types of constraints. The degrees of freedom in rail transport are often lower, but the dependencies are significantly higher.

1.1 Our Contribution

Our main contribution is the integrated modelling of scheduling, routing and energy management constraints in the railway setting. This integration of the various constraints goes beyond existing approaches which are often solved as stages, as discussed in the Problem Statement. Especially the routing is much more relevant and complicated in the railway setting as several constraints such as headway times or deadlock prevention are rather specific and much more rel-

[a] https://orcid.org/0000-0002-7789-8870
[b] https://orcid.org/0000-0003-4936-0018

Grüne, C. and Zieger, S.
Demand-responsive Scheduling in Railway Transportation.
DOI: 10.5220/0011018700003191
In *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2022)*, pages 239-248
ISBN: 978-989-758-573-9; ISSN: 2184-495X

evant.

We develop a model to tackle the overall problem. With this modelling, effects of infrastructure expansion or the size of the vehicle fleet can be quantified by using the modelling as a black box. Thus, different infrastructure variants in combination with the optimal scheduling solution can be used as a decision support tool for the implementation of a demand-responsive railway system.

In the main model, the number of vehicles and requests are fixed and all requests must be served. Further objective variants enable for example maximising the number of passengers served with the available resources. This is relevant when the number of vehicles is fixed and often there are a few requests which have extraordinary costs for fulfilling. Another variation of the objective function is minimising the number of used vehicles while still fulfilling all requests. Here, the service quality is of importance, but the reduction of costs for the acquisition of the vehicles is the main objective.

1.2 Related Work

The Dial-a-Ride Problem (DARP) has a long standing history (Stein, 1978). It is a special routing and scheduling problem in which the users specify requests for their pick-up and drop-off at their origin and destination, respectively. The objective is usually to compute a schedule fulfilling several constraints and maximising the number of transported persons while minimising the costs doing so. Cordeau and Laporte (Cordeau and Laporte, 2003; Cordeau and Laporte, 2007) as well as Ho et al. (Ho et al., 2018) provide an excellent overview on model variants and solution techniques.

In contrast to the also well-studied Periodic Event Scheduling Problem (Serafini and Ukovich, 1989; Liebchen and Möhring, 2007) which has many applications in railway scheduling the DARP is characterised by its irregular requests and corresponding scheduling. The DARP has been studied mainly for rubber-tyred systems. Railway systems offer less degrees of freedom in terms of moving directions for example compared to cars. However, the degree of interdependence is much stronger, e.g. on single track lines (Szpigel, 1973; Landex, 2009). Several scheduling approaches by means of Integer Programming exist for the railway scheduling problem (Castillo et al., 2009; Castillo et al., 2011; Caimi et al., 2017), but their scope is often limited and concern the construction of schedules on main lines or improve dispatching decisions (Weik et al., 2018; Li et al., 2013). Cats and Haverkamp (Cats and Haverkamp, 2018a; Cats and Haverkamp, 2018b) assume an automated demand-repsonsive rail service on main lines and determine the optimal line and station capacity for such a system. To the best of the authors knowledge this is the only related work in this setting.

1.3 Outline

The paper's structure is as follows. In the Preliminaries, the graph representation of the railway infrastructure is presented. Following, the problem is stated informally. In Sec. 3, the main part of the paper, the model is proposed by means of Integer Programming and different variants are discussed. Subsequently, a computational study (Sec. 4) is performed in which first a validation scenario is investigated and afterwards a real railway line are subject to the study. Finally in Sec. 5, the paper is summarised and an outlook on further open work is discussed.

2 PRELIMINARIES

In the beginning, the abstraction from real-world railway infrastructure to a graph is described. The abstraction is necessary to use models which will then be able to provide decision support for real-world instances. In the end of the section the problem is stated informally.

2.1 Railway Network Modelling

The railway network consists of several components and can basically be divided into the line section and stations as denoted in Fig. 1. The stations are characterised by their capacity for dwelling and the possibility of boarding and alighting passengers. The line, in turn, is located between two stations and is usually single or double track. Specifically, in rural areas, the lines are often single track. Depending on the underlying protection system, the lines are often divided into blocks based on the placement of signals. This segmentation is therefore often modelled as several sections of the line. Due to the comparatively high number of individual train movements, which strongly stress the capacity, and the technical possibilities with the use of automated vehicles, a reduction of the block sizes is to be aimed for. In this way, headway times of trains are kept as short as possible and the capacity is used in the best possible way. In fact, the use of automated and connected vehicles likely results in minimum headway times in the realm of the relative braking distance, i.e. just a few seconds for homogeneous vehicles, and is assumed in the following.

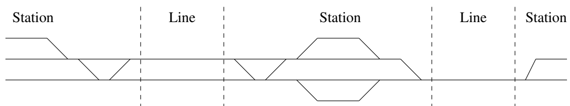


Figure 1: Rail infrastructure with three Stations and a double-track line as well as a single track line between those.

The abstraction of the infrastructure is performed by the introduction of a graph $G=(V,E)$. In Fig. 2 the exemplary transformation for the infrastructure from Fig. 1 is depicted. In this graph, the set of vertices V represents the beginning/end of a block section and especially the station entrance/exit, i.e. switch areas. The edge set E models the block sections and station tracks. Generally the edge set is undirected, but if the direction of movement is relevant, e.g. due to signals on a double track section, the undirected edges can be turned into directed arcs. In case the distance between two stations is longer than one time step, the edge can be subdivided into a path of appropriate length.

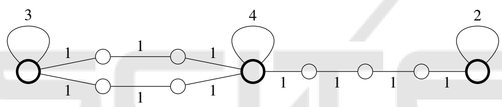


Figure 2: Example mesoscopic infrastructure abstraction by means of a graph with appropriate edge capacities.

The model is focused on the usage of the edges and therefore they contain several properties. First of all, they have a capacity, which is usually one because, for safety reasons, only one train is allowed to be within one section at any given time. We abstract the trains' pre- and post-occupation of sections as they are likely to traverse the network as "moving block". In stations, the edge capacity determines the number of vehicles the station can withhold and can alternatively be modelled as multigraph if all edges should have capacity equal to one. Additionally, edges include the information about the travel time on those. These have to be integer and longer sections are usually split up such that each edge needs one time step to be traversed.

2.2 Problem Statement

Usually, the transport planning problem consists of a series of hard tasks which are often executed one after each other. These are demand estimation, line planning, scheduling, platform and track assignment, rolling stock planning, crew scheduling as well as shunting and maintenance planning (Goossens, 2004). In the case of highly automated unmanned vehicles, the problem reduces to the scheduling, platform and track assignment as well as rolling stock assignment aspects. In fact, the routing of trains of capacity greater than one, which is a sub-problem of this problem, is already NP-hard (Guan, 1998). Thus, this problem is expected not to be solvable efficiently.

The task is to find a feasible schedule transporting the passengers directly from origin to destination while fulfilling all routing and energy management constraints, if possible. The resulting schedule should minimise the costs, i.e. the number of train movements. Other possible objectives are discussed in Sec. 3.2. The passengers specify their origin and destination along with some time window each. When the passengers board or alight, the trains will need to be assigned some dwell time to give sufficient time for doing so. The vehicles operated on the network are constrained by their capacity, i.e. the number of passengers which can be transported at the same time, and the energy capacity. Often, rural railway lines are not electrified and thence the vehicles have to recharge or refuel after some time.

The railway system is characterised by strong interdependencies. One prime example is the minimum headway time, i.e. the least amount of time which has to pass between two successive (or opposing) trains such that the second train can move unhindered from the first train. The trains therefore need to have sufficient spacing between each other.

3 MIXED INTEGER PROGRAMMING FORMULATION

In the following, the problem is to be modelled by means of an Integer Programming formulation. In the first part, an overview on the notation is given. Then, the model is presented and the constraints are explained. Finally, necessary modifications for different objective variants are discussed.

3.1 Model

The relevant notation of the model is summarised in Tab. 1. The table contains the description of the graph, sets, parameters and variables.

Illustrating the use of the tracker variables for the passengers and vehicles, Fig. 3 depicts the states for passenger p assigned to train r. It can be seen that each passenger is always in a clearly defined and dis-

Table 1: Relevant notation for the model.

G	Underlying graph	
$V(G)$	Vertex set of graph G	
$H(G) \subseteq V(G)$	Station vertex set	
$D \in H(G)$	Depot vertex	
$E(G)$	Edge set of graph G	
$H_E(G) \subseteq E(G)$	Station edge (platform) set	
$D_E \in E(G)$	Depot edge	
c_e	Vehicle Capacity of edge $e \in E(G)$	
R	Set of all vehicles	
P	Set of all passengers	
T	Time horizon	
$\overline{T}$	Set of all timestamps of the discretised interval $[0,T]$	
$s(p)$	Start station vertex of passenger $p \in P$	
$s_e(p)$	Start station edge of passenger $p \in P$	
$d(p)$	Destination station vertex of passenger $p \in P$	
$d_e(p)$	Destination station edge of passenger $p \in P$	
$W_s(p)$	Start time window for passenger $p \in P$	
$W_d(p)$	Destination time window for passenger $p \in P$	
w	Dwell time at stations	
$L(p)$	Distance from $s(p)$ to $d(p)$ for $p \in P$	
ε	Percentage distance extension	
$c(r)$	Passenger capacity of vehicle $r \in R$	
$\dot{E}(r)$	Charging speed of vehicle $r \in R$	
$c_E(r)$	Energy storage capacity of vehicle $r \in R$	
$hs^t_p, y^t_p, hd^t_p, ad^t_p$	Auxiliary variables	$\{0,1\}$
$M_{\cdot,\cdot}, W$	Big-M-parameters	
$x^t_{r,e}$	Is vehicle $r \in R$ using edge $e \in E(G)$ at time $t \in \bar{T}$?	$\{0,1\}$
$y^t_{r,p}$	Is person $p \in P$ in vehicle $r \in R$ at time $t \in \bar{T}$?	$\{0,1\}$
$a^t_{h,p}$	Is person $p \in P$ at station $h \in \{s(p), d(p)\} \subseteq H(G)$ at time $t \in \bar{T}$?	$\{0,1\}$
$h^t_{r,h,p}$	Does vehicle $r \in R$ dwell for passenger $p \in P$ at stations $h \in \{s(p), d(p)\} \subseteq H(G)$ at time $t \in \bar{T}$?	$\{0,1\}$
f^t_r	Does vehicle $r \in R$ refuel/recharge at time $t \in \bar{T}$?	$\{0,1\}$
$v_{r,p}$	Does vehicle $r \in R$ transport passenger $p \in P$?	$\{0,1\}$

tinct state. In the beginning, they wait for the train, then they board the train at their origin, travel to their destination and alight there and, finally, their request is completed.

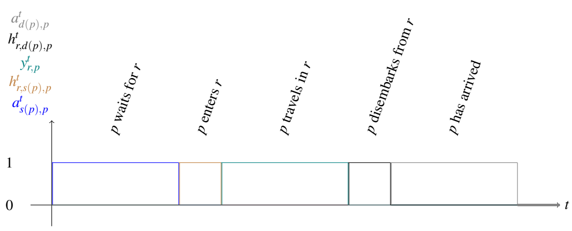


Figure 3: The plot of the 5 tracker variables $a^t_{s(p),p}$, $h^t_{r,s(p),p}$, $y^t_{r,p}$, $h^t_{r,d(p),p}$, $a^t_{d(p),p}$.

Following, the objective and constraints are presented and explained below.

$$\min \sum_{r \in R} \sum_{e \in E \setminus H_E} \sum_{t \in \bar{T}} x^t_{r,e} \quad (1)$$

The objective in Eq. (1) minimises the number of used edges which are not station edges, i.e. travelled distance of all vehicles is minimised. In consequence, the vehicles only move if absolutely necessary and the objective can be understood as the minimisation of operation costs for the operator given the set of vehicles.

$$\sum_{e \in E} x^t_{r,e} = 1 \qquad \forall r \in R, t \in \bar{T} \quad (2)$$

$$\sum_{e' \in N_e[e]} x^{t-1}_{r,e'} \geq x^t_{r,e} \qquad \forall r \in R, e \in E, t \in \bar{T} \setminus \{0\} \quad (3)$$

Eq. (2) ensures that each of the vehicles is located on exactly one edge for each time step. Together with Eq. (3), the vehicles then can only move to adjacent edges and a flow conservation in the time-expanded graph is enforced. The neighbourhood $N_e[e]$ is closed as vehicles have to be able to wait or dwell, e.g. in front of switches or in stations while waiting for the next passenger request.

$$\sum_{r\in R} v_{r,p} = 1 \quad \forall p \in P \quad (4)$$
$$h^t_{r,s(p),p} \leq v_{r,p} \quad \forall r \in R, p \in P, t \in \bar{T} \quad (5)$$
$$y^t_{r,p} \leq v_{r,p} \quad \forall r \in R, p \in P, t \in \bar{T} \quad (6)$$
$$h^t_{r,d(p),p} \leq v_{r,p} \quad \forall r \in R, p \in P, t \in \bar{T} \quad (7)$$

Eq. (4) assigns each passenger to a vehicle such that all passenger requests have to be served. The following three equations (5)-(7) ensure that the tracker variables $h^t_{r,s(p),p}$, $y^t_{r,p}$ and $h^t_{r,d(p),p}$ for each passenger and vehicle pair can only be activated if the passenger is assigned to the vehicle.

$$\sum_{p\in P} y^t_{r,p} \leq c(r) \quad \forall r \in R, t \in \bar{T} \quad (8)$$
$$\sum_{r\in R} x^t_{r,e} \leq c_e \quad \forall e \in E, t \in \bar{T} \quad (9)$$

The next two equations are concerning capacities. Eq. (8) enforces the vehicle's capacity $c(r)$ not to be exceeded, i.e. at most $c(r)$ passengers are able to travel in one vehicle at the same time. The second part limits the line capacity depending on the available infrastructure. In the same time step, only one vehicle is allowed at each line section and therefore c_e is usually set to one for the open line. This constraint ensures a safety distance between two successive trains which is known as the minimum headway distance. On a single track line the constraint furthermore prevents deadlocks, i.e. two trains facing each other without the chance of evasion. In stations, c_e denotes the number of vehicles a station can contain at each step in time.

$$\sum_{t\in W_s(p)} h^t_{r,s(p),p} \geq 1 \quad \forall p \in P \quad (10)$$
$$\sum_{t\in W_d(p)} h^t_{r,d(p),p} \geq 1 \quad \forall p \in P \quad (11)$$

Eqs. (10) and (11) allow the specification of time windows for boarding and alighting for each passenger. With these equations it is possible to map the passengers (dis-)satisfaction with waiting times.

$$\sum_{t\in \bar{T}} y^t_{r,p} \leq (1+\varepsilon)L_p \quad \forall r \in R, p \in P \quad (12)$$

Besides the waiting times, users expect not to take extended detours between their starting point location and destination. Eq. (12) allows to define an accepted detour factor ε extending the length of the shortest path $L(p)$.

$$c_E(r) + \sum_{t'\in[0,t]} (\dot{E}(r) f^{t'}_r - \sum_{e\in E\setminus E_H} x^{t'}_{r,e}) \leq c_E(r)$$
$$\forall r \in R, t \in \bar{T} \quad (13)$$
$$c_E(r) + \sum_{t'\in[0,t]} (\dot{E}(r) f^{t'}_r - \sum_{e\in E\setminus E_H} x^{t'}_{r,e}) \geq 0$$
$$\forall r \in R, t \in \bar{T} \quad (14)$$
$$f^t_r \leq x^t_{r,D_E}$$
$$\forall r \in R, t \in \bar{T} \quad (15)$$

The Eqs. (13)-(15) are concerned with the charging process. Each vehicle starts with a full charge $c_E(r)$. Subsequently, exactly $\dot{E}(r)$ units of energy can be regenerated in each charging step (f). For simplification, only the movement costs one unit of energy. Eq. (13) ensures that charging does not exceed the energy capacity $c_E(r)$ of a vehicle. At the same time, Eq. (13) ensures that there is always enough energy. The last constraint guarantees that the vehicle is at the depot when it is charging.

$$a^0_{s(p),p} = 1 \quad \forall p \in P \quad (16)$$
$$a^T_{s(p),p} = 0 \quad \forall p \in P \quad (17)$$
$$y^0_{r,p} = 0 \quad \forall r \in R, p \in P \quad (18)$$
$$y^T_{r,p} = 0 \quad \forall r \in R, p \in P \quad (19)$$
$$a^0_{d(p),p} = 0 \quad \forall p \in P \quad (20)$$
$$a^T_{d(p),p} = 1 \quad \forall p \in P \quad (21)$$

Eqs. (16)-(21) set the variables which have to be fixed in the beginning or end. In the beginning (Eqs. (16)-(17)), the passengers are definitely waiting at their starting point location $s(p)$ and will not be waiting there at the end of the time horizon. Each passenger p can only be transported in a vehicle r after waiting at the starting point location $s(p)$ and before reaching their destination $d(p)$. Therefore, the y-variable is set to 0 in the beginning and end of the time horizon in Eqs. (18) and (19). Finally, the passengers are never located at their destination $d(p)$ at the beginning of the time horizon, but are definitely there at the end.

$$\sum_{r\in R}\sum_{t\in \bar{T}} h^t_{r,s(p),p} \geq 1 \quad \forall p \in P \quad (22)$$
$$\sum_{r\in R}\sum_{t\in \bar{T}} y^t_{r,p} \geq 1 \quad \forall p \in P \quad (23)$$
$$\sum_{r\in R}\sum_{t\in \bar{T}} h^t_{r,d(p),p} \geq 1 \quad \forall p \in P \quad (24)$$

Each passenger has to be transported. To ensure that all necessary variables are at least enabled once during the time horizon, Eqs. (22)-(24) enforce setting each to one for at least one time step. The a-variables already fulfil this demand as per the previous set of equations.

$$a^t_{s(p),p} + \sum_{r\in R}(h^t_{r,s(p),p} + y^t_{r,p} + h^t_{r,d(p),p}) + a^t_{d(p),p} = 1$$
$$\forall p \in P, t \in \bar{T} \quad (25)$$

$$x^t_{r,s_e(p),p} \geq h^t_{r,s(p),p}$$
$$\forall r \in R, p \in P, t \in \bar{T} \quad (26)$$

$$x^t_{r,d_e(p),p} \geq h^t_{r,d(p),p}$$
$$\forall r \in R, p \in P, t \in \bar{T} \quad (27)$$

$$y^t_{r,p} \leq 1 - a^{t-w}_{s(p),p}$$
$$\forall r \in R, p \in P, t \in \bar{T}, t \geq w \quad (28)$$

$$y^{t-w}_{r,p} \leq 1 - a^{t-w}_{d(p),p}$$
$$\forall r \in R, p \in P, t \in \bar{T}, t \geq w \quad (29)$$

Every passenger has to be in one of the states (waiting at the station before boarding, boarding, driving, alighting, service finished) during the time horizon. Therefore, Eq. (25) enforces one of the corresponding variables always to be set to one. Eqs. (26)-(27) connect the waiting state of the passenger for boarding and alighting with the position of the allocated vehicle. This implies the vehicle to move to the boarding edge, travel with the passenger and stop at the alighting edge as well. The passengers are allowed some time w for boarding and alighting and Eqs. (28)-(29) take these constraints into account.

$$hs^{t-1}_p \leq hs^t_p \quad (30)$$
$$\sum_{r\in R} h^t_{r,s(p),p} \neq 1 + M_{11}h_{d1} \quad (31)$$
$$hs^t_p = 1 + M_{11}(1 - h_{d1}) \quad (32)$$
$$hs^t_p \neq 1 + M_{12}h_{d2} \quad (33)$$
$$a^t_{s(p),p} = 0 + M_{12}(1 - h_{d2}) \quad (34)$$
$$y^{t-1}_p \leq y^t_p \quad (35)$$
$$\sum_{r\in R} y^t_{r,p} \neq 1 + M_{21}y_{d1} \quad (36)$$
$$y^t_p = 1 + M_{21}(1 - y_{d1}) \quad (37)$$
$$y^t_p \neq 1 + M_{22}y_{d2} \quad (38)$$
$$\forall p \in P, t \in \bar{T} \setminus \{0\}$$

$$h^t_{r,s(p),p} = 0 + M_{22}(1 - y_{d2}) \quad (39)$$
$$\forall r \in R, p \in P, t \in \bar{T} \setminus \{0\}$$

$$hd^{t-1}_p \leq hd^t_p \quad (40)$$
$$\sum_{r\in R} h^t_{r,d(p),p} \neq 1 + M_{31}hd_{d1} \quad (41)$$
$$hd^t_p = 1 + M_{31}(1 - hd_{d1}) \quad (42)$$
$$hd^t_p \neq 1 + M_{32}hd_{d2} \quad (43)$$
$$\forall p \in P, t \in \bar{T} \setminus \{0\}$$

$$y^t_{r,p} = 0 + M_{32}(1 - hd_{d2}) \quad (44)$$
$$\forall r \in R, p \in P, t \in \bar{T} \setminus \{0\}$$

$$ad^{t-1}_p \leq ad^t_p \quad (45)$$
$$\sum_{r\in R} a^t_{d(p),p} \neq 1 + M_{41}ad_{d1} \quad (46)$$
$$ad^t_p = 1 + M_{41}(1 - ad_{d1}) \quad (47)$$
$$ad^t_p \neq 1 + M_{42}ad_{d2} \quad (48)$$
$$\forall p \in P, t \in \bar{T} \setminus \{0\}$$

$$h^t_{r,d(p),p} = 0 + M_{42}(1 - ad_{d2}) \quad (49)$$
$$\forall r \in R, p \in P, t \in \bar{T} \setminus \{0\}$$

Eqs. (30)-(49) introduce many auxiliary variables. Each 5-tuple of constraints follows the same pattern. The general idea is to ensure that the passengers are handled in the correct order of variables ($a^t_{s(p),p}, h^t_{r,s(p),p}, y^t_{r,p}, h^t_{r,d(p),p}, a^t_{d(p),p}$). In some cases, e.g. with very loose time windows, the chronological order of the service is not imposed directly. Therefore, these additional constraints need to be incorporated. For the sake of the argument, the general procedure for each block of constraints is discussed by means of the first set as all of those work the exact same way. The blocks are written in non-standard form on purpose for better readability. All transformation steps are done according to Bisschop (Bisschop, 2006, Chap. 7).

First, an auxiliary variable hs^t_p is introduced which ensures that once the corresponding tracker variable $h^t_{r,s(p),p}$ became active the auxiliary variable is always set to one. The following two constraints state that if the corresponding tracker variable $h^t_{r,s(p),p}$ is set to one (A) then the auxiliary variable hs^t_p is set to one as well (B). From this moment on, hs^t_p will be set to one even if $h^t_{r,s(p),p}$ is already set to zero again.

Let $\neg A$ and $\neg B$ denote the negation of A and B as usual. Then the statement "if A then B" is equivalent to the logical expression "(A and $\neg B$) is false". The negation "$\neg$ ($\neg A$ and B)" is then true. This in turn is equivalent to state "($\neg A$ or B) is true" which is represented by Eqs. (31) and (32) using the Big-M-method. The final two equations in this block work the exact same way and state that if the auxiliary variable hs^t_p is set to to one the previous tracker variable $a^t_{s(p),p}$ cannot be one anymore and will not be so, too. This ensures that exactly one tracker variable is set to one for each passenger for each time step.

3.2 Variants

The problem as is can be used in different variants covering different objectives. These also have strong implications on economic factors and passenger satisfaction. In the model presented above, all passengers have to be served and all vehicles could potentially be used, even if unnecessary, as only the number of used edges outside the station is relevant and penalised in the objective function.

First, it might not be possible to serve all request with the available amount of vehicles. Then, the model would just return that there exists no feasible solution. Another approach is to maximise the number of serviced passengers while still minimising the costs for doing so. This goal can be achieved by altering the objective function (Eq. (1)) to

$$\min -W(\sum_{r\in R}\sum_{p\in P} v_{r,p}) + \sum_{r\in R}\sum_{e\in E\setminus H_E}\sum_{t\in \bar{T}} x^t_{r,e}. \quad (50)$$

Some large enough value W ensures that the first part of the sum is always relevant and thus the number of passengers is indeed maximised.

Second, from an economical point of view it makes sense to minimise the number of vehicles in usage. Therefore, one possibility is specifying for example the number of vehicles equal to the number of passengers and then find out how many vehicles are necessary at all. Therefore, another variable $v_r \in \{0,1\}$ is introduced indicating if a vehicle is used. The objective (Eq. (1)) changes to

$$\min W\sum_{r\in R} v_r + \sum_{r\in R}\sum_{e\in E\setminus H_E}\sum_{t\in \bar{T}} x^t_{r,e} \quad (51)$$

and the following constraint is necessary

$$\sum_{p\in P}\frac{v_{r,p}}{|P|} \leq v_r \qquad \forall r\in R. \quad (52)$$

For each vehicle $r \in R$, the number of passengers $(v_{r,p})$ that have used it is counted and the sum is then divided by $|P|$. Since v_r is an integer, the variable is always set to 1 if and only if the vehicle is used and, by the objective function, to 0 otherwise.

4 COMPUTATIONAL STUDY

In this section, at first two scenarios for testing the implementation of the model are described. One of them is a pure validation scenario presenting the conflict management on single track sections. Next, the framework of the implementation is presented. Finally, some results are discussed for the two scenarios which will be the bridge to the conclusion of the paper.

4.1 Scenario Description

In order to evaluate the solving procedure on the mathematical programming formulation two different railway networks are used.

The first instance is a simple artificial infrastructure of two stations with a short single track line in between. This infrastructure is used to perform parameter tests and to validate the correctness of the mathematical programming formulation, especially with regard to conflict detection and avoidance. In Fig. 4 the route map can be found. The numbers correspond to the mileage of the stations.

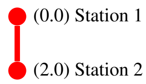


Figure 4: Route map of the validation instance.

Additionally, the solution procedure is tested against a real-world infrastructure as an illustrative example. The former railway line Solingen–Wuppertal-Vohwinkel connected the towns Solingen and Wuppertal, both in North Rhine-Westphalia, Germany. Most of the line is disused, partly even dismantled. The 22 *km* long railway line is mostly single-track and has 10 stations. Fig. 5 presents a the route map for this railway line.

Figure 5: Route map of the railway line Solingen—Wuppertal-Vohwinkel.

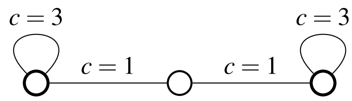
4.2 Framework

The model is the integral part within the procedure which is implemented in Matlab 2019b (MATLAB, 2020) as a frame. There, the infrastructure and request generation is performed. The model itself is written according to the specifications of ZIMPL (Koch, 2005) which is then also used to transform the text model into a machine readable model file.

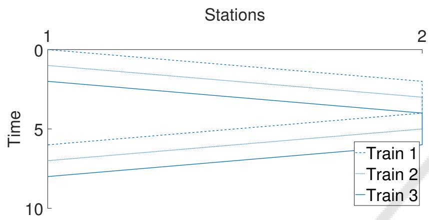
Then, Gurobi (Gurobi Optimization, 2020) is used as a solver and the output is evaluated and visualised within Matlab again. The computations are performed on a machine with an i5-6500@3.2GHz (4 Cores) with 16 GB RAM.

4.3 Results

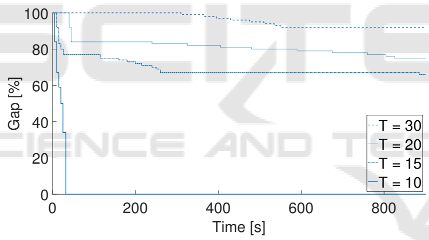
In the first part, some results are presented for the validation scenario and then for the real-world example.



(a) Graph representation for the validation infrastructure.



(b) One optimal schedule for the validation scenario.



(c) Gap over the runtime for different time windows.

Figure 6: Results for the validation scenario.

The abstraction of the route map can be seen in Fig. 6a. Assuming that the vehicles travel at an average speed of about 60 km/h, this results in a distance of about 1 *km* per minute. The minute measure also seems to be a good time length for the time steps in the following cases. In general, there is a trade-off between the accuracy of the modelling and the efficient computability of the solution.

The graph consists of the two stations with self loops (platforms) and an intermediate node that acts as a separator for subsequent train runs. To illustrate, 3 passengers are to travel from one station to the other, respectively, each vehicle can only transport 1 person and the time window is set to 10 minutes and starts at time step 0. In Fig. 6b an optimal solution is plotted as a time-distance diagram. Due to the limited time window, the requests have to be bundled and processed directionally one after the other. This allows the trains to follow each other in the same direction with one time unit between them. If the trains were to run alternately, two time units would be blocked for each train.

In Fig. 6c the gaps between the found solution and the proven lower bound are presented over a maximum computing time of 15 minutes for the same scenario, but with an increasing time window. It can be observed that just by increasing the time window slightly, an optimal solution is found within seconds, but this cannot be effectively proven by the solver using the lower bound. Due to the high number of symmetries and the almost arbitrary pattern according to which the requests can be processed, there are likely problems in quickly proving the optimum. Lower bounds can be helpful here, such as

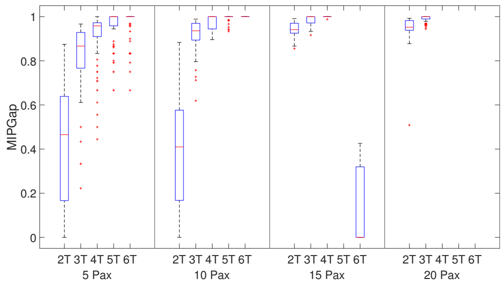
$$\frac{\sum_{p \in P} dist(s_e(p), d_e(p))}{\max_{r \in R} c(r)},$$

which would also have given the value 12 (6 requests with length 2 each) as the correct lower bound for this instance.

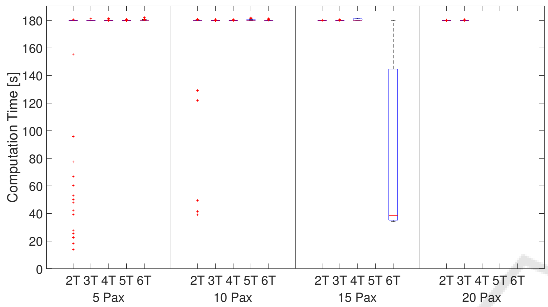
In the second part, the operation on the railway line Solingen–Wuppertal-Vohwinkel is examined as an example. The resulting graph consists of 27 vertices and 36 edges on which the vehicles are routed. The case study is performed with the following set of parameters. The number of passengers ranges between 5 and 20, the vehicles between 2 and 6. A time frame as well maximum ride time of 120 minutes is set and each time step corresponds to one minute which seems appropriate for a reasonable segmentation of this line. The requests are generated randomly with a uniform distribution along the line. Their start time within the first 80 time steps is distributed uniformly as well. Additionally, a time window of 10 minutes allows some flexibility in handling and pooling the requests. The dwell time is set to one minute and the vehicle is able to accommodate 5 passengers. In this scenario charging is not considered to decrease the complexity.

Each of the scenarios is set up 100 times and each trial has a time limit of 3 minutes for the optimisation. A summary of the runs is displayed in Fig. 7

First of all, it must be mentioned that the number of feasible instances or instances for which a feasible solution was found within 3 minutes varies for the different cases. While in the first scenarios a feasible solution could still be found for almost all cases, there are, for example, only 3 feasible solutions for 15 passengers and 6 trains. For three scenarios, no feasible solutions were found or generated at all. In some cases, it is also intuitive that there are fewer



(a) Gap over different passenger (Pax) and train (T) scenarios.



(b) Computation time over different passenger (Pax) and train (T) scenarios.

Figure 7: Results for the computational study on the Solingen–Wuppertal-Vohwinkel railway line.

feasible solutions. For example, it is more difficult to transport 15 passengers with 2 trains rather than with 5 trains, especially if the requests are distributed over the whole line. On the other hand, more trains also create significantly more conflicts, which lead to greater complexity in finding a feasible solution and are also more challenging to coordinate operationally.

In Fig. 7a the relative gap between primal and dual solution of the optimisation problem (MIPGap) is depicted. To begin with, it can be observed that the gap is always less than 1. This seems to indicate an underlying structure of the problem that the solver knows to exploit. Furthermore, with the exception of one outlier (15 passengers, 6 trains), it is evident that the problem size is increasing and that better solutions can be found more quickly for smaller scenarios.

Fig. 7b provides an insight on the actually used computation time. Ignoring the outlier (15 passengers, 6 trains) again, it can be observed that basically only with the smallest train fleet (2 trains) the computation took sometimes less than 3 minutes finding an optimal or close to optimal solution.

The black-box pre-solving already indicates that there seems to be potential for strong pre-processing enabling the processing of larger instances. The determination of the infeasibility of an instance is often reached within the first few seconds, which is crucial because if the timeout were to occur without a solution, the situation with no solution and with a solution not yet found would be indistinguishable.

5 CONCLUSION

Offering demand-responsive transport on rails can increase the attractiveness, availability and accessibility of public transport, especially in rural areas. This necessitates the efficient use of resources, which requires an optimisation of the activities involved. The Integer Program presented in Sec. 3 models the problem with many facets.

Due to the time sensitive encoding of the variables, e.g. the dependence on time steps even in those in which nothing happens the instance size grows very fast. Correspondingly, the computation time is very sensitive on the size of the time horizon. Some constraints, e.g. the enforced correct processing order in Eq. (30)-(49), are not always necessary or even redundant, but in some cases such as very large time windows they are necessary.

All in all, the run time can possibly be decreased by either tuning black-box solvers such Gurobi (Gurobi Optimization, 2020) for this task or developing dedicated tailored algorithms such as by Castillo et al. (Castillo et al., 2009). This approach could then be supported by significant pre-processing. Another possibility is the development of heuristics which provide (ideally provable) good solutions for the problem. Besides the improvement of computation, the problem clearly demands for a robust formulation which should be a logical next step. The variations in the requests over an extended period, e.g. a year, should be considered as uncertain input and an operator should be able to handle different scenarios. The corresponding number of vehicles should be able to satisfy the varying demand. Conclusively, the problem deserves a lot of attention and has much room for improvement in various directions.

ACKNOWLEDGEMENTS

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) — 2236/1.

REFERENCES

Bisschop, J. (2006). *AIMMS optimization modeling*.

Caimi, G., Kroon, L., and Liebchen, C. (2017). Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, 6(4):285–312.

Castillo, E., Gallego, I., Ureña, J. M., and Coronado, J. M. (2009). Timetabling optimization of a single railway track line with sensitivity analysis. *TOP*, 17:256–287.

Castillo, E., Gallego, I., Ureña, J. M., and Coronado, J. M. (2011). Timetabling optimization of a mixed double- and single-tracked railway network. *Applied Mathematical Modelling*.

Cats, O. and Haverkamp, J. (2018a). Optimal infrastructure capacity of automated on-demand rail-bound transit systems. *Transportation Research Part B: Methodological*, 117:378–392.

Cats, O. and Haverkamp, J. (2018b). Strategic planning and prospects of rail-bound demand responsive transit. *Transportation Research Record*, 2672:404–410.

Cordeau, J. F. and Laporte, G. (2003). The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms. *4OR*, 1:89–101.

Cordeau, J. F. and Laporte, G. (2007). The dial-a-ride problem: Models and algorithms. *Annals of Operations Research*, 153:29–46.

European Commission (2020). Transforming europe's rail system.

Goossens, J.-W. (2004). Models and algorithms for railway line planning problems.

Guan, D. J. (1998). Routing a vehicle of capacity greater than one. *Discrete Applied Mathematics*, 81(1):41 – 57.

Gurobi Optimization (2020). Gurobi optimizer reference manual.

Ho, S. C., Szeto, W. Y., Kuo, Y.-H., Leung, J. M. Y., Petering, M., and Tou, T. W. H. (2018). A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, 111:395–421.

Koch, T. (2005). Rapid mathematical programming.

Landex, A. (2009). Evaluation of railway networks with single track operation using the uic 406 capacity method. *Networks and Spatial Economics*, 9(1):7–23.

Li, F., Gao, Z., Li, K., and Wang, D. Z. (2013). Train routing model and algorithm combined with train scheduling. *Journal of Transportation Engineering*, 139:81–91.

Liebchen, C. and Möhring, R. H. (2007). The modeling power of the periodic event scheduling problem: railway timetables—and beyond.

MATLAB (2020). *version 9.7.0 (R2019b)*. The MathWorks Inc.

Schindler, C. and von Stillfried, A. (2020). Fahren auf Sicht - Ein Betriebskonzept für den fahrerlosen Nahverkehr. *Eisenbahntechnische Rundschau*, 69(10):22–27.

Schlaht, J., Frink, L., Laumen, P., Pfeifer, A., Schindler, C., and Nießen, N. (2018). Automated Nano Transport System - Ansatz zur Entwicklung autonomer Schienenfahrzeuge. In *Proceedings of the 1st International Railway Symposium Aachen*, pages 60–77.

Serafini, P. and Ukovich, W. (1989). A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2:550–581.

Stein, D. M. (1978). Scheduling dial-a-ride transportation systems. *Transportation Science*, 12:232–249.

Szpigel, B. (1973). Optimal train scheduling on a single line railway.

Weik, N., Zieger, S., and Nießen, N. (2018). Traffic management heuristics for bidirectional segments on double-track railway lines. In *Operations Research Proceedings 2017*, pages 729–735. Springer.