ECONSTOR Make Your Publications Visible.

A Service of

ZBW

Leibniz-Informationszentrum Wirtschaft Leibniz Information Centre for Economics

Weicker, Karsten

Article — Published Version Optimisation of seat reservations on trains to minimise transfer distances

Operational Research

Provided in Cooperation with: Springer Nature

Suggested Citation: Weicker, Karsten (2023) : Optimisation of seat reservations on trains to minimise transfer distances, Operational Research, ISSN 1866-1505, Springer, Berlin, Heidelberg, Vol. 23, Iss. 3,

https://doi.org/10.1007/s12351-023-00786-6

This Version is available at: https://hdl.handle.net/10419/308110

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.



WWW.ECONSTOR.EU

https://creativecommons.org/licenses/by/4.0/

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



ORIGINAL PAPER



Optimisation of seat reservations on trains to minimise transfer distances

Karsten Weicker¹

Received: 7 February 2022 / Revised: 7 March 2023 / Accepted: 12 June 2023 / Published online: 17 July 2023 © The Author(s) 2023

Abstract

This paper introduces a novel optimisation problem motivated by the real-world application of transferring between trains. The driving idea is to minimise the walking distances of all transferring passengers by optimising the assignment of passengers to railcarriages in the seat reservation process. The focus of this work is on formalising and modelling the problem, which has not been studied before. It aims to provide a framework for future considerations. We present three versions of the problem with increasing difficulty: one-train-one-station, one-train-many-stations, and a general problem with multiple trains and stations. Since the simplest version of the problem is a pure assignment problem in the form of minimum-cost bipartite matching, our problem modelling and algorithmic solution approach remain closely related to maxflow problems as they represent a proven method for assignment problems. However, even the one-train-many-stations problem cannot be solved by transforming it into a standard maximum-flow problem. The main result shown is the NP-hardness of the general problem.

Keywords Seat reservation \cdot Maximum flow problem \cdot Bipartite matching \cdot NP-hardness

Mathematics Subject Classification $05C90 \cdot 68Q17 \cdot 90B20 \cdot 90B80 \cdot 90C27 \cdot 90C35$

1 Introduction

The operation of public transport depends on diverse, often operation-critical optimisation problems with different planning scopes (Huisman et al. 2005)—from strategic goals such as line planning to short-term goals such as crew scheduling (Neufeld et al.

Karsten Weicker karsten.weicker@htwk-leipzig.de

¹ Faculty of Computer Science and Media, HTWK Leipzig University of Applied Sciences, Karl-Liebknecht-Straße 132, 04277 Leipzig, Germany

2021). For an individual traveller, many framework conditions such as line planning cannot be influenced. All the more important are the optimisation problems which affect the solution of his individual journey within the given framework conditions. These optimisation problems include the search for train connections (Pyrga et al. 2007), which, in addition to a pure search for the fastest connection, also takes into account other factors such as the reliability of a connection or the walking distances to be covered in urban public transport (Disser et al. 2008). Seat reservation is also one of the customer-oriented optimisation problems, although up to now this has primarily been considered for group reservations as a variant of the 2-dimensional knapsack problem (Clausen et al. 2010).

This paper approaches the seat reservation process from a different angle. In order to increase travel comfort, passengers' seat reservations should be made in such a way that individual walking distances are as short as possible. This applies in particular to transfers at stations where, due to clumsy reservations in two trains, the entire platform has to be walked although both trains stop there. When transferring to a train at the same platform, the next reservation should be close to the position where you left the train before. When transferring to a train at another platform, or even at the beginning and end of the journey, the reserved seats should be close to the platform access. The objective of this work is to develop a mathematical model of the application problem and to classify the optimisation problems in terms of problem complexity. To our knowledge, this problem has not yet been formally addressed.

In the long term, this should answer questions as to whether there are suitable heuristics that can be used to efficiently find an acceptable approximate solution offline. Based on this, an experimental study can give an answer as to how big a difference between such a reservation system and a random placement of passengers is. Last but not least, an optimised reservation system could also have an impact on the operation of the railway, since transfer times between connections and dwell times of trains at the station can be planned differently.

These considerations are largely driven by the future vision that tickets will not only be used digitally via smartphone, but that the seats reserved will be determined shortly before the journey. In particular, this allows us to take into account situations such as trains arriving in reverse, missing railcarriages, and replacement trains. An online adjustment of reservations during operation is out of the scope of this work.

In the following section, the newly considered problem is framed by a literature review. Section 4 introduces the basic concepts of the railway infrastructure. The following sections define the optimisation problem with increasing problem difficulty: Sect. 5 considers one train in one station, Sect. 6 one complete train, and Sect. 7 a complete interconnection network. The latter problem is then analysed in Sect. 8 with regard to its computational complexity. The final Sect. 9 summarises the findings.

2 Related work

Seat reservation on trains has been the subject of various studies and publications in the past, which we summarise in this section. In addition, we also discuss some research that is closely related to the results presented here. Online seat reservations. The oldest publications on seat reservations take an economic perspective and examine online reservation algorithms where not all requests are yet complete when seats are allocated. Those publications examine the degree to which online reservation algorithms can fill trains to capacity (Boyar and Larsen 1999; Bach et al. 2000; Miyazaki and Okamoto 2010), the so-called accommodating ratio. Two pricing schemes are compared: a fixed flat price and costs that are proportional to the length of the journey. In short, these works show that a significant proportion of seats cannot be allocated by an online algorithm because each decision is made with incomplete information. This effect can be countered by certain techniques, e.g. reserving only partial sections or by requiring passengers to change seats on a train (Solanki and Patidar 2019).

Offline group seat reservations. The conceptual counterpart is the offline allocation of seat reservations, where all reservation requests are known. From an algorithmic point of view, this is hardly interesting for individual reservations. Therefore, the research here deals with group reservations, which make it difficult to allocate all seats. Clausen et al. (2010) transform the problem into a two-dimensional knapsack problem or two-dimensional bin packing and investigate branch-and-bound algorithms with different bounds. Deplano et al. (2019) have extended the problem by attaching additional profits to various properties of the seats.

Offline seating arrangements. While the train utilisation is significantly better with an offline reservation, the business model of a railway company suggests an online algorithm. Kohrt and Larsen (2005) address this conflict by separating the mere decision as to whether a reservation can be fulfilled (as an online problem) from the actual allocation of seats (as an offline problem). They present a data structure in which the insertion and deletion of a reservation is possible in $O(\log p)$ (where p is the number of stations). This concept is an essential basis for the problem presented in our article, which attempts to achieve greater customer-friendliness when allocating seats offline in the second step.

Dwell times of passengers and the transfer coordination design problem. When approaches are presented in the literature to make a passenger's journey more comfortable, it is mostly about adjustments to departure schedules and better coordination of the different trains. Liu et al. (2021) present an extensive survey article on this type of problem, which mostly has minimising transfer waiting time, minimising total cost, or maximising of the number of successful transfers as the optimisation objective. Since these considerations concern mass transportation, they are only of marginal importance for our problem. Mostly, cycle times, frequencies, or departure times are optimised, e.g. Jansen et al. (2002) minimise the weighted sum of transfer times with a tabu search heuristic and Nieuwenhuis (2021) uses a metaheuristic to adjust the departure times by few minutes achieving a 5.7% reduction in transfer times. A few works also consider passenger assignment as a decision variable, but this refers to the choice between different possible connections or paths and not to seat reservations (Zhou et al. 2019; Chu et al. 2019).

Modelling of connecting trains. In the reservation problems from the literature, transfer processes have not been taken into account so far, which is why the associated problem models do not have the concept of the connecting train. Therefore, we will now take a brief look at the models used in timetable query. Pyrga et al.

(2007) present two different ways of introducing time-dependencies into the graph representing the rail network and train connections. The time-dependent model contains only one node for each station and, for many queries, is superior to the time-expanded model, which duplicates the station nodes in the graph for the different departure times. To make the time-expanded model more competitive, Delling et al. (2009) simplified the original model by representing unimportant stations differently. As part of our work, we decided not to store time in the graph or model itself, but to consider time-dependencies externally when generating passenger travel plans.

Transfer cost. Passenger transfer costs beyond the dwell time between two connecting trains are rarely considered in the existing literature. An example of this is work by Li et al. (2021) in the context of optimised coordination of full-length and short-turn trains on an urban train line. The transfer costs are levied rather coarsely on the basis of the number of transfers.

3 Problem description

In this section, the optimisation problem is introduced informally and all assumptions concerning the modelling of the optimisation problem are summarised.

3.1 Transfer cost

In this paper we assume that the routing of the trains, the train schedule and the trains used are invariant constant quantities. Likewise, the set of passengers with their individual travel plans, i.e. the assignment to a sequence of trains, is an input variable for our problem.

This leaves the seat reservations for passengers as the only decision variables within this optimisation problem. The objective variable is the length of the travellers' walking distances when transferring between the trains as well as when boarding and alighting the train at the beginning and at the end of the journey. The distances travelled by all passengers should be kept as short as possible, which is why square distances are used in the rest of the paper.

This problem is introduced in three variants with increasing difficulty. First, only boarding a train at a station is considered. In the second variant, the course of a train is modelled along the stations of its routing, the passengers stay on the train for an individual section at a time and they board the train and leave the train at a specific position on the platform in each case. In the last variant, a complete network of trains is considered and the passengers' journey can extend over several trains.

3.2 Assumptions for the optimisation problem

This section summarises the constraints and assumptions that we use to keep the model of the optimisation problem manageable enough for the scope of this paper.

The focus of the work exclusively questions the extent to which clever seat reservations can reduce the distances travelled. All other factors are assumed to be invariant and we abstract from too fine details in the model.

Assumption 1 We assume that the assignment of a track to a train is immutable and a direct consequence of the given rail network.

Assumption 2 We assume that each railcarriage has exactly one entrance. Without loss of generality, the entrance is positioned in the middle of the railcarriage.

For purely technical reasons, we assume that the train line can be determined from the number of a railcarriage. In reality, a railcarriage is used several times a day on different train lines.

Assumption 3 Each railcarriage is only used in one train on one train line.

A model that takes into account individual travellers could easily be provided with in-depth details such as train departure times, waiting times, and alternative connecting trains. This would directly imply that route planning is also done as part of the optimisation. As this is outside the focus of the problem considered here, we refrain from providing this information and assume that the individual travel plans of the passengers were preceded by corresponding considerations.

Assumption 4 We assume that the temporal coupling of trains at a station arises from the existing individual travel plans.

In the literature, seat reservations have so far been considered mainly in the context of group reservations, without which reservations would be a simple problem in itself (Clausen et al. 2010). In this introductory paper, we do not consider group reservations because we first want to measure the complexity of the problem that arises from considering transfer paths alone.

Assumption 5 We assume that only individual reservations are possible.

This limits the directly applicable practical value of the considerations presented in this paper. Nevertheless, group reservations could be considered in a practical implementation. One possible approach could be direct reservation for groups and downstream optimisation for individual travellers.

4 Modelling the infrastructure

This section introduces the concepts of rail infrastructure, in particular stations, train routes, and trains deployed. The operating conditions at the station can then be used to define the transfer costs for a passenger.

4.1 Notations

In order to keep the model clear and to increase readability, the notations are summarised compactly in three tables. Table 1 contains the sets, indices, and parameters. Table 2 shows the relations and functions used to access certain properties of the objects in the model. Table 3 contains the single decision variable that is used in all three variants of the problem.

4.2 Station configuration

Definition 1 (Station) A station *st* consists of several platforms $st.pl \subset \mathbb{N}$, each of which usually has two tracks. Each platform $pl \in st.pl$ has positions $pl.pos \subset \mathbb{N}$ that correspond to sections of the length of a railcarriage and that are labelled by a continuous sequence of numbers. In addition, the station has a cross connection between

	Semitton of sets, indices, and parameters
Notation	Explanation
ST	Set of all stations
st	Parameter ($\in ST$) denoting one station
st.pl	Set of all platforms at station st
pl	Parameter denoting one platform
st.cross	Position ($\in \mathbb{N}$) of the access to all platforms at station <i>st</i>
pos	Parameter $(\in \mathbb{N})$ denoting a relative position at any platform
$T\mathcal{R}$	Set of all train routes
tr	Parameter ($\in TR$) denoting one train route
tr.len	Number ($\in \mathbb{N}$) of stations visited in train route <i>tr</i>
$tr.st_i$	<i>i</i> -th station ($\in ST$) of train route $tr (1 \le i \le tr.len)$
\mathcal{W}	Set of all possible railcarriages
\mathcal{W}_{tr}	Set of railcarriages ($\subseteq W$) in train <i>tr</i>
<i>w</i> , <i>w</i> _{<i>i</i>}	Parameters ($\in W$) denoting a railcarriage
w.tr	Train route ($\in \mathbb{TR}$) on which railcarriage <i>w</i> is used
w.no	Sequence number $(\in \mathbb{N})$ of railcarriage <i>w</i> in the deployed train for <i>w</i> .tr
w.cap	Seat capacity $(\in \mathbb{N})$ of railcarriage <i>w</i>
\mathcal{P}	Set of all passengers
р	Parameter ($\in \mathcal{P}$) denoting a passenger
$p.\mathrm{pl}_\mathrm{in}$	Platform ($\in \mathbb{N}$) at which passenger <i>p</i> arrives (in the one-station problem and the one-train problem)
p.pl _{out}	Platform $(\in \mathbb{N})$ from which passenger <i>p</i> leaves (in the one-train problem)
<i>p</i> .pos _{in}	Position $(\in \mathbb{N})$ at which passenger <i>p</i> arrives at the platform (in the one-station problem and the one-train problem)
p.st _{in}	Station ($\in ST$) at which passenger <i>p</i> boards (in the one-train problem)
p.st _{out}	Station ($\in ST$) at which passenger <i>p</i> alights (in the one-train problem)
p.route	Individual travel route [$\in (ST \times TR)^+ \times ST$) of passenger <i>p</i> (in the general problem)]

 Table 1
 Definition of sets, indices, and parameters

 Table 2
 Relations and functions

	cost	Determines the square distance of two positions at two platforms
	train	Assigns a sequence of railcarriages to a train route
	pl	Assigns a platform of a station to a train route
	pos	Assigns a position at the platform to the train route
	dir	Indicates the direction of travel at the platform for the train route
	cpos	Computes the position of a railcarriage at the platform from the operation conditions <i>pl</i> , <i>pos</i> , and <i>dir</i>
	осс	Function used in the one-station problem to parameterise the number of already occupied seats in each railcar- riage
Table 3Definition of decisionvariables	Notation	Explanation
	Мар	Assignment of each passenger to a railcarriage of each involved train

Notation Explanation





the platforms at position *st*.cross $\in \mathbb{N}$ which is also the only access to the platforms. ST denotes the set of all possible stations.

Example 1 Figure 1 shows an exemplary station with two platforms. From platform 1 you can board the trains on tracks 1 and 2. At position 3, there is a cross connection to access the platform. Platform 2 illustrates the freedoms of the model: The platform does not start at position 1, but still has the cross connection at the same position as platform 1. In addition, platform 2 serves three tracks. A train on track 4 can only stop at positions 8, 9, and 10.

In this paper, only transfer paths resulting from seat reservations on one or more trains are considered and minimised. This means that unavoidable path sections of the transfer are not taken into account when calculating the costs. Because in the static view of the paper, due to Assumption 1, the track of each train stop is unchangeable, the distance between two platforms cannot be minimised and is not included in the subsequent definition of the costs.

We also do not take into account the distance within a railcarriage to the actual seat, but are only interested in the routes to the entrance of the carriage (Assumption 2).

In order to achieve the lowest possible transfer costs for all travellers, we define the transfer costs as squared distances. As we will later consider the sum of all transfer costs, the squared cost function should penalise very long journeys by individual travellers—a technique borrowed from linear regression (cf. Downey 2014, p. 118).

Definition 2 (Avoidable distance cost when changing trains) A passenger who has to walk from position *pos* on platform *pl* to position *pos'* on platform *pl'* at station *st* has the avoidable squared distance cost

$$cost(pl, pos, pl', pos') = \begin{cases} (pos - pos')^2, & \text{if } pl = pl' \\ (\|pos - st. cross\| + \|pos' - st. cross\|)^2, & \text{otherwise} \end{cases}.$$
(1)

Because we only take into account the distance travelled on the platforms and not between the platforms, the platform numbers only serve as identifiers. The further definitions do not require a consistency check regarding the number of platforms at a station or the length of the platforms.

Example 2 Figure 2 illustrates two scenarios for calculating distance costs. In scenario (a), the trains stop at the same platform and there is a cost of $(2 - 5)^2 = 9$. In scenario (b), the trains stop at different platforms. In that case, only the horizontal distances are taken into account for the calculation of the distance costs. This leads to the costs $(||4 - 3|| + ||5 - 3||)^2 = 3^2 = 9$.

4.3 Train configuration

To introduce the trains that connect the different stations, we distinguish between two different concepts. We first introduce the pure train route as a sequence of the stations visited. This is initially detached from the physical train, which is addressed as the second concept in Definition 5. The train route is also often referred to as a train line, which usually means the abstraction of several trains serving the same sequence of stations at different times. The definition of the train route implicitly includes a timetable and can therefore only be implemented through one train per day.

Definition 3 (Train route) A train route tr is defined by its timetable with a sequence of tr.len stations

$$\langle tr.st_1, \dots, tr.st_{tr.len} \rangle \in ST^*$$
 (2)



(a) Example for two trains at the (b) Example for two trains at difsame platform. ferent platforms.

Fig. 2 Two scenarios to illustrate the computation of the distance cost. The horizontal distances are valid and the vertical distances are disregarded

with $tr.st_i \neq tr.st_j$ for $i \neq j$. A train route tr defines a partial order $<_{tr}$ on the set ST as follows:

$$tr.st_i < tr.st_j \text{ if and only if } i < j$$

and $tr.st_i \le tr.st_i \text{ if and only if } i \le j.$ (3)

Let $T\mathcal{R}$ be the set of all considered train routes. For the sake of simplicity, $S\mathcal{T}_{tr} \subseteq S\mathcal{T}$ denotes the set $\{tr.st_1, \ldots, tr.st_{tr.len}\}$.

The notation using the Kleene star in ST^* denotes the set of all sequences consisting of an arbitrary number of elements in ST.

The train actually used on a train route is introduced in the following definition by its railcarriages with their seating capacities. The separation of the train from the train route makes it possible to easily replace a train with a differently sized train in later work.

Definition 4 (Deployed train) A train route tr is implemented by a deployed train. For this purpose, a concrete sequence of railcarriages is assigned to the train

$$train: T\mathcal{R} \to \mathcal{W}^*$$
$$tr \mapsto \langle w_1, \dots, w_{\ell} \rangle \quad (w_i \neq w_j \text{ for } i \neq j)$$
(4)

with the set W containing all possible railcarriages. For the sake of simplicity, $W_{tr} \subseteq W$ denotes the set $\{w_1, \dots, w_\ell\}$.

Because of Assumption 3, for $tr, tr' \in T\mathcal{R}$ with $tr \neq tr'$ it holds: $train(tr) \cap train(tr') = \emptyset$. As a consequence, the following definition introduces the properties of a railcarriage within a train.

$$w.tr = tr$$
 where $w \in \mathcal{W}_{tr}$ (5)

and the number of a railcarriage w within tr is denoted by

w.no = i where
$$train(w.tr) = \langle w_1, \dots, w_\ell \rangle$$
 and $w = w_i$. (6)

Moreover, each railcarriage *w* has a capacity of *w*.cap $\in \mathbb{N}_0$ seats.

Example 3 On the German railways (*Deutsche Bahn*), typical railcarriages are the ICE open-plan coach with 80 seats, the ICE coach with combined open-plan and compartment seating with a capacity of 72 seats, and the Intercity 2 double-decker coach with a capacity of 112 seats. The railcarriage with the on-board restaurant does not contain any reservable seats and therefore has a capacity of 0.

The presented definition of the deployed train affects the reservations on all sections of the entire train route. In contrast, the operating condition at the station in the following definition only affects the position of the individual railcarriages in the station and thus only the reservations of the passengers changing trains there.

Definition 6 (Operation at station) A train route *tr* deployed as *train*(*tr*) is operated at station $s \in ST_{tr}$

- On platform $pl(tr, s) \in \mathbb{N}$,
- With the smallest position $pos(tr, s) \in \mathbb{N}$ that is used by train(tr), and
- The direction of travel *dir*(*tr*, *s*) ∈ { /, / } where / indicates an ascending order of the railcarriages on the platform and / indicates a descending order.

Example 4 In the complex and closely interwoven rail network of Europe, the operation of a train and the order of its railcarriages is not trivial. Figure 3 shows a simple example with three stations. While the railcarriages at stations A and B are arranged with the direction of travel (\checkmark), the train changes its direction of travel at station B, arriving at station C in reverse order (\nearrow). However, if the train has to take the detour shown in dashed lines, it will keep its original direction at station C as well.

Caution: The direction of travel is a concept that refers exclusively to one station of the train. So this direction of travel actually depends on the side from which a train enters the station and the physical orientation of the train. In the following examples we will simplify and derive the direction of travel solely from the physical orientation.

The following definition determines how the positions of the railcarriages are actually calculated from the operation at the station.



Fig. 3 Small example with three stations to illustrate the operation at the station. The dashed line corresponds to a train route that can be used as a reroute

Definition 7 (Carriage position) A train route *tr* using *train(tr)* with railcarriages W_{tr} stops at station *st* according to position p = pos(tr, st) and direction d = dir(tr, st). Then, when boarding and alighting, a railcarriage $w \in W_{tr}$ stops at platform position

$$cpos(w, p, d) = \begin{cases} p + w.no - 1, & \text{if } d = \nearrow \\ p + \ell - w.no, & \text{otherwise.} \end{cases}$$
(7)

Example 5 The resulting position of the railcarriages is shown in Fig. 4 for two trains. Depending on the variables *pos* and *dir*, the four cars are positioned differently on the platform.

This completes the introduction of the necessary infrastructure. The deployed train and its operation at the station provide all the information needed to calculate the distance cost for a passenger who leaves a railcarriage and goes to the exit (at the connecting corridor of the station).

5 One-station transfer problem

This section introduces the *one-station transfer problem* (or one-train-one-station problem) as a first, simple version of the transfer problem. It only considers multiple passengers boarding one train at one station.

5.1 Optimisation problem

Each passenger has a fixed point of arrival, either at the entrance to the platform or at an arbitrary position when actually transferring from another train. In addition, only selected seats are still available on the train in question. Now an allocation of the



passengers to the free seats in the railcarriages is to be determined for which the sum of the distance costs is minimal.

Definition 8 (One-station transfer problem) A train route *tr* is implemented by $train(tr) = \langle w_1, ..., w_\ell \rangle$ and operated at station $st \in ST_{tr}$ according to $pl^* = pl(tr, st)$, $pos^* = pos(tr, st)$ and $dir^* = dir(tr, st)$. In each railcarriage w_j $(1 \le j \le \ell)$, there are $occ(w_j) \le w_j$.cap seats permanently occupied. Given is the set of passengers \mathcal{P} where each $p \in \mathcal{P}$ arrives at platform $p.pl_{in} \in \mathbb{N}$ at position $p.pos_{in} \in \mathbb{N}$ $(1 \le i \le n)$ and transfers to train route *tr*. Then, in the one-station transfer problem, an assignment

$$Map: \mathcal{P} \to \mathcal{W}_{tr} \tag{8}$$

is sought such that

$$\forall w \in \mathcal{W}_{tr} : occ(w) + ||\{p \in \mathcal{P} \mid Map(p) = w\}|| \le w.\text{cap and}$$

$$\sum_{p \in \mathcal{P}} cost(p.\text{pl}_{in}, p.\text{pos}_{in}, pl^*, cpos(Map(p), pos^*, dir^*))$$
(9)

is minimal over all possible mappings Map.

Example 6 A small problem instance is shown in Fig. 5a, where there are three railcarriages with a capacity of 10 seats. Up to seven seats are already occupied in the individual railcarriages. There are 10 arriving passengers. A solution is shown in Fig. 5b with minimal $\cot 7 \cdot 1^2 + 1 \cdot 2^2 + 2 \cdot 0 = 11$.

5.2 Runtime complexity and problem difficulty

Because there is a close link between assignment problems and minimum flow problems (Ford and Fulkerson 1956; Gabow and Tarjan 1988; Kennedy 1995), this paper uses network flow as a modelling and solution approach for the problems presented. The simple problem variant described in this section is equivalent to weighted bipartite matching.

Example 7 Figure 6 shows how the transfer problem of Fig. 5 can be modelled by an minimum-cost flow problem (MCFP). There is a vertex for each position at the platform in which the number of arriving passengers are induced using the incoming



Fig. 5 Small example for the one-station transfer problem

edges. Analogously, there is a vertex for each railcarriage and the free seat capacity of the railcarriage is bound by the capacity of the outgoing edge. The edges between the positions and the railcarriages determine the transfer cost for all possible transfer distances. The minimum cost flow corresponds to the solution in Fig. 5b.

Because there are known polynomial-time algorithms for weighted bipartite matching, the following corollary holds.

Corollary 1 *The one-station transfer problem is in* **P***.*

This was shown for minimum-cost bipartite matching by Kuhn (1955) via the Hungarian method with runtime $\mathcal{O}(n^4)$ (where *n* is the number of vertices). Gabow and Tarjan (1988) presented a maximum-flow algorithm with runtime $\mathcal{O}(\sqrt{nm}\log(n \cdot C))$, where in addition *m* is the number of edges and *C* bounds the absolute value of the integer cost values. Schwartz et al. (2005) presented an approximation algorithm that solves the problem in $\mathcal{O}(n^2)$ with high probability. It is likewise an open question whether further properties of the problem instances can be used to design more efficient exact algorithms for the one-station transfer problem.



Fig. 6 Transformation of the one-station transfer problem in Fig. 5 into a minimum-cost flow problem (MCFP). The edge labels denote (*capacity*, *cost*) and the underlines capacity values are lower bounds for the required flow

6 One-train transfer problem

In this section, the problem is generalised by considering a train with all stops. Each passenger enters the train at one station and leaves it at a later stop. The problem is referred to as *one-train transfer problem* (or one-train-many-stations problem).

6.1 Optimisation problem

The following definition introduces the extended problem. Besides additional information for the passengers and some technical conditions, the costs for the footpaths of all passengers at all stations are summed up as a major change.

Definition 9 (One-train transfer problem) A train route *tr* serving the stations $\langle tr.st_1, \ldots, tr.st_{tr.len} \rangle$ is implemented by $train(tr) = \langle w_1, \ldots, w_\ell \rangle$ and operated at stations ST_{tr} according to the functions *pl* (platforms), *pos* (positions) and *dir* (directions).

Given is the set of passengers \mathcal{P} where each passenger $p \in \mathcal{P}$ is travelling using train route *tr*

- from station $p.st_{in} \in ST_{tr}$ at platform $p.pl_{in} \in \mathbb{N}$ and position $p.pos_{in} \in \mathbb{N}$
- to station $p.st_{out} \in ST_{tr}$ at platform $p.pl_{out} \in \mathbb{N}$ and position $p.pos_{out} \in \mathbb{N}$

where $p.st_{in} <_{tr} p.st_{out}$.

Then, in the one-train transfer problem, an assignment

$$Map: \mathcal{P} \to \mathcal{W}_{tr}$$

is sought such that

$$\forall st \in \mathcal{ST}_{tr} \ \forall w \in \mathcal{W}_{tr} : \| \{ p \in \mathcal{P} | Map(p) = w \land p.st_{in} \leq_{tr} st \land st <_{tr} p.st_{out} \} \| \leq w.cap \text{ and}$$
(10)

$$\sum_{p \in \mathcal{P}} (cost(p.pl_{in}, p.pos_{in}, pl(tr, p.st_{in}), cpos^{*}(p, tr, p.st_{in})) + cost(p.pl_{out}, p.pos_{out}, pl(tr, p.st_{out}), cpos^{*}(p, tr, p.st_{out})))$$
with $cpos^{*}(p, tr, st) := cpos(Map(p), pos(tr, st), dir(tr, st))$
(11)

is minimal over all possible mappings Map.

Example 8 We consider a small, highly simplified example on the train route of Fig. 3 with three stations. The train consists of three carriages with two seats each. The upper part of Fig. 7 shows the sections where seven passengers arrive at the platform and leave the platform. Inside the station boxes the placement and travel

direction of the train is shown as well as one solution for the assignment of the passengers to the railcarriages.

The passenger's distances going from station A to station B equals a cost of 1 when boarding and a cost of 0 when alighting. Passengers from A to C have a cost of $2 \cdot 1$ when boarding and a cost of $2 \cdot 2^2 + 1$ when alighting. Passengers from B to C have a cost of $2 \cdot 1 + 2^2$ when boarding and a cost of $2 \cdot 1$ when alighting. This brings the total cost to 20.

The passenger from A to B could also be placed in the first railcarriage at the same cost. However, moving one of the three passengers travelling from A to C from the third railcarriage to one of the other railcarriages would significantly increase the cost of boarding the passengers at station B. This would also increase the total cost: If only one passenger is placed in the third railcarriage at station A, the cost would be at least 22.

6.2 Algorithmic approach using maxflow

Since the single transfer at a station corresponds to weighted bipartite matching, it makes sense to model the one-train transfer problem in principle as a minimum-cost flow problem (MCFP) using a sequence of matching problems. Figure 8 illustrates the basic idea:

- Passengers transferring at a station are modelled by complete bipartite subgraphs, similar to the example in Fig. 6 for the one-station transfer problem. In Fig. 8, these are the incoming edges to station A and the outgoing edges from station B.
- The movement of the train between two stations is modelled by edges that determine the new position for each railcarriage. In Fig. 8, these are the edges between station A and station B.

However, Fig. 8 shows that such naïve modelling is not sufficient: Because the passengers are required to exit at station B in reverse order (see the left part of Fig. 8), the correct minimum cost solution has transfer cost 2. But when the two



Fig. 7 Small example for the one-train transfer problem: Given problem instance with a capacity of 2 in each railcarriage. The arrows within the station boxes indicate a solution with minimal transfer cost



Fig.8 Counterexample for reducing the one-train transfer problem to minimum-cost flow problem (MCFP). The left part shows the transfer problem and the right part the respecting model using two subsequent one-station problems. The first number at the edges corresponds to the capacity. The second number to the cost per transfered unit. The incoming and outgoing edges have a lower capacity bound noted as $\underline{1}$

stations are lined up as subsequent matching problems, the direct assignment of the individual passenger to the boarding and alighting position is lost. As a consequence the sketched solution in the right part of Fig. 8 is computed that has the minimal transfer cost 0. The naïve modelling as MCFP only ensures that the correct number of passengers enters and leaves the train.

Instead, the modelling must be extended to include differently labelled items, leading to an instance of the minimum-cost multicommodity flow problem (MCMCF) (Ahuja et al. 1993). A separate commodity is used for each combination of traveller origin and destination.

Example 9 Figure 9 shows an instance of the minimum-cost multicommodity flow problem that can be used to solve the small example with three stations given in Fig. 7.

We note that the one train transfer problem has more complexity than the mere sequencing of one station transfer problems. This also means that we cannot yet offer a polynomial-time algorithm for the one-train transfer problem, since MCMCF is known to be NP-complete (Even et al. 1976). Besides the open research question whether the one-train transfer problem still lies in \mathbf{P} , efficient heuristics and approximation algorithms are also a short-term research goal.

7 General train transfer problem

In this section, the problem is extended again by combining different trains. The problem is referred to as *general train transfer problem*.



Fig. 9 Minimum-cost multicommodity flow problem equivalent to the one-train transfer problem in Fig. 7. The edges are labeled with (*capacity*, *cost*) and the nodes are labeled with the maximal capacity. The incoming and outgoing edges have a lower capacity bound noted as $\underline{1}$

7.1 Optimisation problem

To generalise the problem, the following definition introduces several trains and the individual travel plans of each passenger. As with the result of a timetable query, each passenger uses a combination of individual journey sections of the trains and thus determines the transfer between the trains.

Definition 10 (General train transfer problem) Given is the set of train routes TR implemented according to *train* and operated in the stations in ST according to *pl* (platforms), *pos* (positions) and *dir* (directions). The set P describes all passengers where each $p \in P$ is travelling with an individual travel schedule *p*.route of length *p*.len:

$$p.\text{route} = (p.\text{st}_1, p.\text{tr}_1, \dots, p.\text{st}_{p.\text{len}}) \in (ST \times TR)^+ \times ST$$
(12)

where $p.st_i, p.st_{i+1} \in ST_{p.tr_i}$ and $p.st_i <_{p.tr_i} p.st_{i+1}$ for all $1 \le i < p.$ len.

Then, in the general train transfer problem, a partial function

$$Map: \mathcal{P} \times \mathcal{TR} \to \mathcal{W} \tag{13}$$

defined for all (p, tr) with $p \in \mathcal{P}$ and $tr \in \{p.tr_1, \dots, p.tr_{p.len-1}\}$ is sought such that

$$\forall tr \in T\mathcal{R} \ \forall st \in S\mathcal{T}_{tr} \ \forall w \in \mathcal{W}_{tr} : \| \{ p \in \mathcal{P} \mid \exists 1 \le i < p.len : p.tr_i = tr \land Map(p, tr) = w \land p.st_i \le_{tr} st \land st <_{tr} p.st_{i+1} \} \| \le w.cap \text{ and}$$
(14)

$$\sum_{p \in \mathcal{P}} \left(cost(0, p.st_1, p.st_1.cross, pl(p.tr_1, p.st_1), cpos^*(p, p.tr_1, p.st_1)) + \sum_{1 < i < p.len} cost(pl(p.tr_{i-1}, p.st_i), cpos^*(p, p.tr_{i-1}, p.st_i), pl(p.tr_i, p.st_i), cpos^*(p, p.tr_i, p.st_i)) + cost(pl(p.tr_{p.len}, p.st_{p.len}), cpos^*(p, p.tr_{p.len}, p.st_{p.len}), 0, p.st_{p.len}.cross) \right)$$

with $cpos^*(p, tr, st) := cpos(Map(p, tr), pos(tr, st), dir(tr, st))$ (15)

is minimal over all possible mappings Map.

In the general problem, we assume that passengers always enter the system at the cross connection. Therefore, platform 0 is used for all arriving passengers and the position of the cross connection $p.st_1.cross$ is used as the position. The same applies to passengers leaving the system at the destination station.

Unlike time-expanded graphs for travel connection queries (Pyrga et al. 2007), we do not specify departure times at all and only link trains by travelling passengers (Assumption 4).

This property enables the modelling of time anomalies: After changing trains several times, we arrive back at the first train at the beginning of our journey similar to the time loop in the movie "Groundhog Day". However, all examples below (including instances generated from other problems) do not exhibit such time anomalies.

Example 10 Figure 10 presents a small connection network with three trains and eight stations. Train t_1 serves stations A, B, C, D, and E and has two railcarriages with a capacity of 3. Train t_2 serves stations F, G, D, and H and has three railcarriages with a capacity of 2. And train t_3 travels directly from G to B and has one railcarriage with a capacity of 3. Table 4 introduces a total of 13 travellers with their individual travel plans.



Fig. 10 Train network as part of an example for the general train transfer problem

passengers	route	passengers	route
a b c_1, c_2, c_3 d	$B \xrightarrow{t_1} E$ $A \xrightarrow{t_1} D$ $A \xrightarrow{t_1} D \xrightarrow{t_2} H$ $F \xrightarrow{t_2} G \xrightarrow{t_3} B \xrightarrow{t_1} C$	$\overline{e_1,e_2,e_3}$ f_1,f_2 g h	$\begin{array}{c} \mathbf{G} \xrightarrow{t_2} \mathbf{D} \xrightarrow{t_1} \mathbf{E} \\ \mathbf{F} \xrightarrow{t_2} \mathbf{H} \\ \mathbf{F} \xrightarrow{t_2} \mathbf{D} \\ \mathbf{C} \xrightarrow{t_1} \mathbf{E} \end{array}$

 Table 4
 Passengers as part of the general train transfer problem given in Fig. 10

A possible solution with minimal transfer costs is shown in Fig. 11. The aim is to keep individual contributions to boarding, transferring, or alighting as small as possible and to avoid the values $2^2 = 4$ or $3^2 = 9$. In the given solution, cost 4 occurs when passengers e_1 , f_1 , and h disembark; they are unavoidable because, due to the railcarriage capacities, passengers have to be placed at higher costs. The solution has total cost of 33.

7.2 Solution using max flow

Let us assume here that the general problem should also be solved via a maximum flow problem. In principle, an MCMCF could be used as in Sect. 6.2, where each commodity corresponds to one passenger. However, the difficulty arises that a flow does not have to follow the specified route of a passenger's individual travel plan. Instead, the maximum flow is used to reproduce the timetable query, since there are several paths in the graph that connect a starting point and an ending point via



Fig. 11 One possible solution for the instance of the general train transfer problem given in Fig. 10 and Table 4. The labelling on the edges shows the occupancy of the railcarriages. The transfer costs for boarding, transferring and alighting are shown at the nodes

different trains. However, the problem to be solved requires that the routes of the passengers be implemented directly.

As a countermeasure, we need to expand the MCMCF with individual lower capacity bounds for each commodity (cf. Cappanera and Frangioni 2003). This way, the basic path can be described for each passenger.

We note here that while this is a fundamentally viable path for possible optimisation, we do not pursue this idea further. Modelling with the tight restrictions leads to overloaded maximum flow problems, which calls efficiency into question—especially since the theorem in the following section rather suggests the use of metaheuristics. Nevertheless, other optimisation techniques, e.g. mixed-integer programming solvers, might be able to efficiently solve problem instances of moderate size or with special properties of real-world scenarios.

8 Computational complexity

To assign the general train transfer problem to a problem class of computational complexity, we first transform it into a decision problem.

Definition 11 (Decision variant of the transfer problem) An instance of the general train transfer problem (TR, ST, train, pl, pos, dir, P) can be turned into a decision problem by adding an upper bound $cost_{max}$ for the minimal transfer costs.

This leads us to the following statement, which suggests that there is no algorithm for solving the problem in polynomial runtime.

Theorem 2 (NP-hardness of the general problem) *The decision variant of the general train transfer problem is NP-hard*.

The proof of the theorem is developed in the following sections.

8.1 Idea of the reduction

A common proof technique for the NP-hardness of a problem A is the reduction of a known NP-hard problem B to problem A. This means that from the detailed information of a problem instance of B a problem instance of A is calculated, so that from its solution the solution of the problem instance of B can be inferred. The conversion itself, however, may require at most polynomial time.

We prove the theorem by constructing a polynomial-time reduction from 3-SAT, the satisfiability of a formula in conjunctive normal form, to the general train transfer problem. However, we restrict the considered space of admissible formulas with the following assumptions. Assumption 6 Each clause has at least two literals.

Assumption 7 All literals in a clause use different variables.

Without loss of generality, only formulas are considered that have clauses with two or three literals. Furthermore, a variable may appear only once in each clause. Clauses with only one literal imply the truth value of the associated variable and the formula can be simplified (cf. Hromkovič 2004, p. 185).

In short, we translate the propositional 3-SAT-formula into a train route network, where the clauses are assigned to stations and the left-to-right reading corresponds to the travelling direction of the involved trains. Simplified, the logical variables correspond to passengers boarding, transferring or terminating their journey at the stations that contain the variable in their associated clauses. For each clause (or its corresponding station), a train is set up to transport the up to three variables of its literals to their respective next clause/station: clause c_k is included as a stop for the train starting at clause c_i (i < k) if there is a common variable in c_i and c_k that does not occur in clauses $c_i + 1, \ldots, c_k - 1$. For each variable considered in this construction, the corresponding passenger's itinerary requires the use of the trains thus defined.

A minimum-cost solution to the transfer problem must correspond to an assignment of the logical variables to truth values that satisfies the 3-SAT problem. Therefore, each deployed train contains a railcarriage that carries the variables whose assignment satisfies the clause c_i corresponding to the first station of the train. An additional railcarriage carries the remaining variables that do not satisfy the clause. Depending on the size *s* of the clause c_i , the capacity of the railcarriage with the satisfying variables is set to *s* and that of the railcarriage with the non-satisfying variables is set to *s* – 1. Consequently, at least one variable must be placed in the satisfying railcarriage. If a variable changing to another clause due to its travel plan is negated in the next clause, the direction of travel at the station is inverted: consequently, a variable satisfying the previous clause is classified as a non-satisfying variable in the next train route (and vice versa) with transfer cost of 0. If several negated and non-negated variables enter the next clause, additional precautions must be taken, which are discussed in the next section.

Furthermore, the same costs (amounting to 1) are incurred for each passenger when entering the system, regardless of whether the respecting variable is assigned the logical value true or false. The same applies when leaving the system.

Thus, satisfactory assignment to the logical variables of the 3-SAT formula corresponds to an occupancy of the seats where only the costs for the beginning and the end of each trip of a variable are incurred, i.e. the cost 2-(number of variables).

If the minimum-cost solution to the train transfer problem serves all passengers and has higher costs, the truth value of one of the variables changes during the train journey and there is no satisfying assignment for the 3-SAT formula.

8.2 Reduction algorithm

Algorithm 1 presents the individual steps of the reduction. The technical details are explained in more depth in the rest of this section.

For each clause, a train with three railcarriages is introduced in step 2(a). The middle railcarriage has capacity of 0 and is positioned at each station at the cross connection where passengers arrive and leave the station. The first railcarriage takes the satisfying variables, the last railcarriage the non-satisfying ones. In a clause with three (resp. two) literals, the first railcarriage has 3 (resp. 2) seats and the last carriage has a capacity of 2 (resp. 1).

Since checking whether there is a satisfying literal in each clause corresponds to the seat occupancy during the train's journey, the corresponding train must travel a distance. Therefore, for the *i*-th clause, a station S'_i is introduced next to the train route's starting station S_i , which serves as the train's first stop [steps 1(a) and 2(b)]. If the *i*-th clause is the first clause containing a variable, the respecting passenger boards the train in station S_i . If the *i*-th clause is the last clause containing a variable, the respecting passenger alights the train in station S'_i [step 2(b)].

All trains stop at the same platform and the same positions within the stations, aligned with each other. As a consequence there are no costs for changing between the first railcarriages in the same direction of travel; analogously for changing between the last railcarriages.

Figure 12 illustrates how the route of the train is constructed from a clause in step 2(b). Since the variables involved in clause 3 are negated (related to the last clause visited), the train must change its direction of travel [step 2(c)]. Variable \bar{x}_3 is also negated in clause 4, but since the train has already changed to the negated state before, the direction of travel must not be inverted again. A renewed change of direction would be necessary if clause 4 would say x_3 instead of \bar{x}_3 .

Obviously, this construction is not sufficient when both negated and non-negated literals occur in the next clause because the inversion of train affects all involved variables. To solve this problem, we introduce an additional train [steps 2(d)-2(f)]. Let *j* be the current clause and *k* be the next clause. We introduce a new train that also starts from station *j'* to station *k*, but arrives there in the opposite direction. Figure 13 shows this technique for the variables x_1 and x_2 , of which only x_2 is negated in the next clause. At the additionally introduced station 1', the passenger belonging to x_2 boards the newly introduced train. Thus x_1 and x_2 arrive at different positions at station 2, although they were assigned to the same railcarriage in clause 1. The new train has the same capacity restrictions as the original train; however, this does not imply any restriction as only passengers of the original train will transfer to the new train. Tables 5 and 6 shows for all possibles cases how the train routes and the additional train are introduced.

Algorithm 1 Reduction of a 3-SAT instance to an instance of the general train transfer problem.

- **3SAT-to-Traintransfer**(formula *F*) computes a set of stations ST, a set of train routes TR, their implementation *train*, and a set of passengers P in such a way that the solubility of the train transfer problem indicates the solubility of the 3-SAT formula. Let $F \equiv c_1 \land ... \land c_k$ be a formula with *k* clauses and literals from the set of variables *V*. According to assumption 6, the *i*-th clause c_i has the form $(\ell_{i,1} \lor \ell_{i,2})$ or $(\ell_{i,1} \lor \ell_{i,2} \lor \ell_{i,3})$.
- 1. (initialisation of stations and passengers)
 - (a) For each clause in F two stations are generated: $ST = \{S_i, S'_i \mid 1 \le i \le k\}$
 - (b) For each variable $x \in V$: create a passenger p_x , search the first clause c_i where x is used, and initialise the travel route p_x .route = (S_i) . Set $\mathcal{P} = \{p_x \mid x \in V\}$.
 - (c) Iterate through the clauses and, for each variable *x*, create a sequence $clauses_x$ that contains the clauses with *x* or \overline{x} as a literal.
 - (d) Initialise $T\mathcal{R} = \emptyset$ and *train* accordingly.
- 2. (train routes and passenger routes.) For each clause c_i :
 - (a) In the remainder of the formula, search those clauses in which the variables of clause c_i reappear for the first time (using *clauses_x*). The result are three clauses c_q , c_r , and c_s (i < q < r < s), two clauses c_q and c_r (i < q < r), one clause c_q (i < q) or no clause.
 - (b) Establish the train route tr_i with stations S_i , S'_i , S_q , S'_q , S_r , S'_s , S_s , S'_s according to the clauses in the previous step. The case with no clause results in S_i and S'_i only, with one clause in S_i ..., S'_q , etc. Add tr_i to TR. Modify *train* such that tr_i has three railcarriages with capacities (3,0,2), if c_i has three literals, and capacities (2,0,1) otherwise.
 - (c) (if c_i has only one variable in common with each other clause.) Set the direction of travel at each station to \nearrow if the literal in the respective clause is equal to the one in c_i , and to \checkmark otherwise. For each variable x used in c_i look up the subsequent entry c_j in $clauses_x$. If there is an entry c_j append (tr_i, S_i) ($j \in \{q, r, s\}$) to p_x .route. If c_i is the last entry in $clauses_x$, append (tr_i, S'_i) instead.
 - (d) (if c_i and c_q have more than one variable in common and neither none nor all of the variables in the literal of c_q are negated relative to c_i .) Introduce a train route tr'_i with stations S'_i and S_q . Add tr'_i to $T\mathcal{R}$. Set the railcarriages of tr'_i and their capacities in *train* to (3,0,3).
 - (e) analogously to (d) with r instead of q.
 - (f) (if step (c) was not executed) Update the direction of travel and the passenger routes according to Tables 5 and 6.
- 3. (return.) ST, TR, train, P

Exactly one passenger is created for each variable (step 1(b)) and the trains of the itinerary are composed according to the construction of the trains explained above (steps 2(c) and 2(f)). For each passenger, his last station is the intermediate station of the last clause in which the according variable occurs. Each passenger enters the station of his first clause at the middle position of the railcarriages and leaves the last station also at the middle position. Tables 5 and 6 shows the resulting passenger routes for the complex cases in step 2(f).

8.3 Proof

The proof of Theorem 2 follows directly from the two lemmata discussed in this section, which determine the correctness and algorithmic runtime complexity of the reduction.



Fig. 12 Example visualizing the construction of a train from the first 3-SAT clause



Fig. 13 Example for the introduction of an additional train. Both x_1 and x_2 re-appear in the next clause but only x_2 is negated. Therefore, the passenger for x_2 will be routed using the additional lower train, and x_1 will stay in the original train

Lemma 3 Algorithm 1 performs a correct reduction from 3-SAT to the general transfer problem.

Proof The construction of an instance of the train transfer problem (ST, TR, train, P) from a given 3-SAT instance F with variables V was presented in detail in the previous section.

Suppose *F* is satisfiable and $\mathcal{A} : V \to \{\text{true, false}\}\)$ is a satisfying assignment. Then we show by contradiction that there is also a corresponding train reservation with the minimum cost 2||V||. So we assume that the minimum cost solution requires a transfer cost > 2||V||. This is only the case if a passenger does not reboard at exactly the same position at a stopover, which means that the associated variable changes its truth value between two clauses. However, this case is only possible if all passengers want to get from a railcarriage with capacity 3 (or 2) to a railcarriage with capacity 2 (or 1) due to different directions of travel. However, this means that the clause corresponding to the station is not satisfied with the initial fulfilling assignment, which implies that the entire formula is not satisfied. Yet, there is a satisfying assignment \mathcal{A} and, due to the construction in the reduction algorithm, we know that in each clause the literals evaluated true by \mathcal{A} can be placed in the

Table 5	Different cases for the	he processing of a	a clause c_i for	which exactly	two variables	reappear for the
first tim	e in the same clause	in the remainder	of the formula	a. The clause c	i_i contains eith	er literals $\ell, \ell',$
and ℓ'' (in any order) or only	literals ℓ' and ℓ''				

Literal in clause ¹		Train route tr_i^2		Add. train	Add. train Append to passenger rou			
c_q	Cr	$\overline{S_i}$	S_q	S_r	tr'_i to	$p_{var(\ell)}$	$p_{var(\ell')}$	$p_{var(\ell'')}$
l	ℓ',ℓ''	7	7	1	_	(tr_i, S_q)	(tr_i, S_r)	(tr_i, S_r)
$\overline{\ell}$	ℓ',ℓ''	1	1	1	—	(tr_i, S_q)	(tr_i, S_r)	(tr_i, S_r)
l	$\overline{\ell'}, \overline{\ell''}$	1	1	2	_	(tr_i, S_q)	(tr_i, S_r)	(tr_i, S_r)
$\overline{\ell}$	$\overline{\ell'}, \overline{\ell''}$	1	1	2		(tr_i, S_q)	(tr_i, S_r)	(tr_i, S_r)
l	$\ell', \overline{\ell''}$	1	1	1	$S_r \swarrow$	(tr_i, S_q)	(tr_i, S_r)	(tr_i, S'_i, tr'_i, S_r)
$\overline{\ell}$	$\overline{\ell'}, \ell''$	1	1	1	$S_r \nearrow$	(tr_i, S_q)	(tr_i, S_r)	(tr_i, S'_i, tr'_i, S_r)
ℓ',ℓ''	l	1	1	1	_	(tr_i, S_r)	(tr_i, S_q)	(tr_i, S_q)
ℓ',ℓ''	$\overline{\ell}$	1	1	1	—	(tr_i, S_r)	(tr_i, S_q)	(tr_i, S_q)
$\overline{\ell'}, \overline{\ell''}$	l	1	1	1	—	(tr_i, S_r)	(tr_i, S_q)	(tr_i, S_q)
$\overline{\ell'}, \overline{\ell''}$	$\overline{\ell}$	1	1	1		(tr_i, S_r)	(tr_i, S_q)	(tr_i, S_q)
$\ell', \overline{\ell''}$	l	1	1	1	$S_q \swarrow$	(tr_i, S_r)	(tr_i, S_q)	(tr_i, S'_i, tr'_i, S_q)
$\overline{\ell'}, \ell''$	$\overline{\ell}$	1	1	1	$S_q \nearrow$	(tr_i, S_r)	(tr_i, S_q)	(tr_i, S'_i, tr'_i, S_q)
ℓ',ℓ''	3	1	1		_		(tr_i, S_q)	(tr_i, S_q)
$\overline{\ell'}, \overline{\ell''}$	3	1	1		—		(tr_i, S_q)	(tr_i, S_q)
$\ell', \overline{\ell''}$	3	1	1		$S_q \swarrow$		(tr_i, S_q)	$(tr_i, S_i^\prime, tr_i^\prime, S_q)$

¹The variables of clause c_i reappear for the first time in the clauses c_q and c_r (i < q < r). The table entries indicate whether the literal is negated.

²The train route is specified in the column header and the table entries indicate the direction of travel at each station.

³The variable of literal ℓ does not reappear in any clause or the clause c_i contains only two literals.

 4 If c_i is the last clause of $\ell,$ append the primed version of the last station entry. Analogously for ℓ' and ℓ''

Negation of literals in clause	Train route tr_i^{1}	Add. train	Append	to passer	nger route ²
c_q	$\overline{S_i S_q}$	tr'_i to	$p_{var(\ell)}$	$p_{var(\ell')}$	$p_{var(\ell'')}$
ℓ,ℓ',ℓ''	11	_	(tr_i, S_q)	(tr_i, S_q)	(tr_i, S_q)
$\ell, \ell', \overline{\ell''}$	11	$S_q \swarrow$	(tr_i, S_q)	(tr_i, S_q)	(tr_i, S'_i, tr'_i, S_q)
$\overline{\ell}, \overline{\ell'}, \ell''$	11	$S_q \nearrow$	(tr_i, S_q)	(tr_i, S_q)	(tr_i, S_i', tr_i', S_q)
$\overline{\ell}, \overline{\ell'}, \overline{\ell''}$	11	_	(tr_i, S_q)	(tr_i, S_q)	(tr_i, S_q)

¹The train route is specified in the column header and the table entries indicate the direction of travel at each station.

²If c_i is the last clause of ℓ , append the primed version of the last station entry. Analogously for ℓ' and ℓ''

Table 6	Different cases for
the proc	essing of a clause c_i
with lite	erals ℓ, ℓ' , and ℓ'' (in
any orde	er) if all three variables
reappea	r in the same clause

associated first railcarriage. Because of this contradiction, the assumption must have been wrong.

Conversely, let the solution of the train transfer problem have transfer cost 2||V||. Assume that there is no satisfying assignment A for the formula F. Again, we know from the construction of the reduction algorithm that missing transfer costs in the changeover mean that the truth value of the variables involved remains consistent. In addition, there must be a satisfying literal in each clause due to the capacities of the railcarriages. This contradicts the assumption, which must therefore also be false. This proves the statement of the lemma.

Lemma 4 The runtime complexity of Algorithm 1 is in P.

Proof In the runtime analysis we use of the following simple facts: All clauses are bound by length 3 ($\in \Theta(1)$) and the number of clauses ||C|| in *F* is bound by ||F|| as given by the equation $\frac{1}{3} \cdot ||F|| \le ||C|| \le ||F||$.

In the initialisation phase, the runtime of step (a) is in $\mathcal{O}(||F||)$, step (b) in $\mathcal{O}(||V|| \cdot ||F||)$, step (c) in $\mathcal{O}(||V||)$, and step (d) in $\Theta(1)$. In the second phase, totalled over all clauses, steps (b) can be accomplished in $\mathcal{O}(||F||)$, all other steps in $\mathcal{O}(||F||^2)$.

Since $||V|| \leq ||F||$, the total runtime is in $\mathcal{O}(||F||^2)$ and, thus, the reduction algorithm in P.

8.4 Examples

Finally, the polynomial reduction of 3-SAT to the general train transfer problem will be illustrated with two small examples.

Example 11 Consider the following small propositional formula in 3-CNF with variables $\{x_1, x_2, x_3\}$ and four clauses.

$$\overbrace{(x_1 \lor \overline{x}_2 \lor x_3)}^{\text{clause 1}} \land \overbrace{(\overline{x}_1 \lor x_2)}^{\text{clause 2}} \land \overbrace{(\overline{x}_1 \lor \overline{x}_2 \lor \overline{x}_3)}^{\text{clause 3}} \land \overbrace{(x_1 \lor \overline{x}_3)}^{\text{clause 4}}$$
(16)

The conversion of the 3-SAT-formula to the general train transfer problem results in the problem instance shown in Fig. 14.

There are two solutions for this problem instance that are shown in Table 7. All the transfer cost between the trains are 0. And, therefore, the total cost is 6 for board-ing the first train and leaving the trains at stations 3b and 4b.

Solution 1 corresponds to the satisfying assignment x_1 , x_2 , $x_3 = 0$ for formula (16): x_2 is seated in the first railcarriage of train 1, which means that it satisfies clause 1. Being a negated literal in clause 1, x_2 corresponds to the truth value 0 (false). Both x_1 and x_3 are placed in the third railcarriage of train 1 and are non-negated literals in clause 1. Consequently, both variables have the truth value 0 (false) as well, since they do not satisfy the clause.

Solution 2 corresponds to the satisfying assignment x_1 , $x_2 = 1$ and $x_3 = 0$.



Fig. 14 Resulting problem instance for formula (16). The individual itineraries are given by the labels at the sections of the trains

Table 7	Solutions	for the	problem	instance	in	Fig.	14
---------	-----------	---------	---------	----------	----	------	----

Solution	Train 1	Train 2	Train 2a	Train 3	Train 3a	Train 4
1	$x_2 x_1, x_3$	<i>x</i> ₁ <i>x</i> ₂	<i>x</i> ₂	$x_1, x_2, x_3 $	<i>x</i> ₁	$x_3 x_1$
2	$x_1 \mid \mid x_2, x_3$	$x_2 x_1$	<i>x</i> ₂	$x_3 \mid \mid x_1, x_2$	<i>x</i> ₁	$x_1, x_3 $

There are two solutions for the given problem. The initial seat occupancy of the individual carriages is displayed in the fields of the table, with the railcarriages separated by

Example 12 We consider a small propositional formula in 3-CNF with variables $\{x_1, x_2\}$ and four clauses. This is a minimal example for a formula that cannot be satisfied.

$$\overbrace{(x_1 \lor x_2)}^{\text{clause 1}} \land \overbrace{(x_1 \lor \overline{x_2})}^{\text{clause 2}} \land \overbrace{(\overline{x_1} \lor \overline{x_2})}^{\text{clause 3}} \land \overbrace{(\overline{x_1} \lor x_2)}^{\text{clause 4}} (17)$$

Figure 15 shows the resulting instance of the the general train transfer problem.

There are only solutions with the minimum cost 8, which is composed of the cost 4 for the start and end of the journeys and the cost 4 for a transfer to a carriage at another position on the platform. All solutions are shown in Table 8. The first four solutions meet the capacity limits of the railcarriages as long as this is possible; a transfer operation with cost $\neq 0$ is only carried out if the capacity of a railcarriage would be exceeded. The passengers placed in this way are written in bold.

The solutions in the lower part of Table 8 already perform an expensive, non-positional placement of a passenger at an earlier stage. All these solutions are variations of the first four solutions.



Fig. 15 Resulting problem instance for formula (17). The individual itineraries are given by the labels at the sections of the trains

		1	8				
Solution	Train 1	Train 1a	Train 2	Train 2a	Train 3	Train 3a	Train 4
1	$x_1, x_2 $	<i>x</i> ₂	$x_1 x_2$	<i>x</i> ₁	$x_1 x_2$	<i>x</i> ₂	$x_1, x_2 $
2	$x_1 x_2$	$ x_2$	$x_1, x_2 \mid \mid$	<i>x</i> ₁	$x_2 x_1$	<i>x</i> ₂	$x_1 x_2$
3	$x_1 x_2$	$ x_2$	$x_1, x_2 \mid \mid$	<i>x</i> ₁	$x_2 x_1$	<i>x</i> ₂	$x_2 x_1$
4	$x_2 x_1$	<i>x</i> ₂	$x_2 x_1$	$ x_1$	$x_1, x_2 \mid \mid$	<i>x</i> ₂	$x_1 x_2$
1'	$x_1, x_2 \mid \mid$	<i>x</i> ₂	$x_1 x_2$	x ₁	$x_1 x_2$	$ x_2$	$x_1, x_2 \mid \mid$
2'	$x_1 x_2$	$ x_2$	$x_1, x_2 \mid \mid$	<i>x</i> ₁	$\mathbf{x_1}, x_2 \mid \mid$	<i>x</i> ₂	$x_1 x_2$
2"	$x_1 x_2$	$ x_2$	$x_1, x_2 \mid \mid$	x ₁	$x_1, x_2 \mid \mid$	<i>x</i> ₂	$x_1 x_2$
2""	$x_1 x_2$	$ x_2$	$x_2 \mathbf{x_1}$	$ x_1$	$x_1, x_2 \mid \mid$	<i>x</i> ₂	$x_1 x_2$
3'	$x_1 x_2$	$ x_2$	$x_1, x_2 \mid \mid$	<i>x</i> ₁	$x_2 x_1$	x ₂	$x_2 x_1$
4'	$x_2 x_1$	x ₂	$x_2 x_1$	$ x_1$	$x_1, x_2 \mid \mid$	<i>x</i> ₂	$x_1 \mid \mid x_2$

 Table 8 Solutions for the problem instance in Fig. 15

There are ten solutions for the given problem with transfer cost 8. The initial seat occupancy of the individual carriages is displayed in the fields of the table, with the railcarriages separated by . The transfer causing the internal cost of 4 is marked in bold face

There is no solution to this problem with cost 2 (number of variables)= 4. Consequently, the given instance of the decision problem with $cost_{max} = 4$ cannot be satisfied. This corresponds to the unsatisfiability of formula (17).

9 Conclusions

This paper introduces a new, previously unexplored optimisation problem for train connections with seats to be reserved, which minimises walking distances when transferring between trains. With the increasing degree of digitisation, these considerations could soon reach a relevance threshold for real-world application.

The train transfer problem is defined in three variants which reflect the radius of operation. In principle, all three problem variants can be mapped to minimumcost maximum flow problems with different extensions. However, while the simplest problem belongs to the complexity class P, NP-hardness is shown for the general problem in this paper.

While this work deals exclusively with the modelling of the new optimisation problem and its theoretical classification, it opens up a large number of research questions to be investigated in the future. This includes the development of heuristics and approximation algorithms, the identification of benchmark problems, and possibly the development of realistic problem generators that produce problem instances tunable by difficulty. The primary goal is to determine through in-depth research and empirical studies to what extent such problem cases can be solved on a larger scale. Furthermore, possible metaheuristics raise the question of the costbenefit ratio: How much computing time must be invested to generate acceptable approximations, and are these costs worthwhile compared to the benefits?

In a vision where reservations are taken via online procedures and then customerfriendly seats are assigned offline just before departure, the basic problem needs to be expanded. On the one hand, the problem must also be extended to group reservations and it must be examined up to which group size or number of group reservations such a two-stage procedure is still feasible. On the other hand, one of the strength of the method is the ability to react quickly to unfamiliar situations. In order to realise this, a non-stationary variant of the problem must be introduced, in which a change in the operating conditions leads to an online optimisation of the seat assignment that was initially optimised offline.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest We have no conflict of interest to disclose that could have appeared to influence the work reported in this paper.

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

References

- Ahuja RK, Magnanti TL, Orlin JB (1993) Network Flows: theory, algorithms, and applications. Prentice Hall, Englewood Cliffs
- Bach E, Boyar J, Jiang T, et al (2000) Better bounds on the accommodating ratio for the seat reservation problem. In: Du D, Eades P, Estivill-Castro V, et al (eds) Computing and combinatorics, 6th annual international conference, COCOON 2000. Springer, pp 221–231
- Boyar J, Larsen KS (1999) The seat reservation problem. Algorithmica 25(4):403-417
- Cappanera P, Frangioni A (2003) Symmetric and asymmetric parallelization of a cost-decomposition algorithm for multicommodity flow problems. INFORMS J Comput 15(4):369–384
- Chu JC, Korsesthakam K, Hsu YT (2019) Models and a solution algorithm for planning transfer synchronization of bus timetables. Trans Res Part E Log Trans Rev 131:247–266
- Clausen T, Hjorth AN, Nielsen M et al (2010) The off-line group seat reservation problem. Eur J Oper Res 207(3):1244–1253
- Delling D, Pajor T, Wagner D (2009) Engineering time-expanded graphs for faster timetable information. In: Ahuja RK, Möhring RH, Zaroliagis CD (eds) Robust and online large-scale optimization: models and techniques for transportation systems. Springer, pp 182–206
- Deplano I, Yazdani D, Nguyen TT (2019) The offline group seat reservation knapsack problem with profit on seats. IEEE Access 7:152358–152367
- Disser Y, Müller-Hannemann M, Schnee M (2008) Multi-criteria shortest paths in time-dependent train networks. In: McGeoch CC (ed) Experimental algorithms: 7th international workshop, WEA 2008. Springer, Berlin, pp 347–361
- Downey AB (2014) Think stats: exploratory data analysis, 2nd edn. O'Reilly, Sebastopol
- Even S, Itai A, Shamir A (1976) On the complexity of timetable and multicommodity flow problems. SIAM J Comput 5(4):691–703
- Ford LR, Fulkerson DR (1956) Maximal flow through a network. Can J Math 8:399-404
- Gabow HN, Tarjan RE (1988) Almost-optimum speed-ups of algorithms for bipartite matching and related problems. In: Simon J (ed) Proceedings of the 20th annual ACM symposium on theory of computing. ACM, pp 514–527
- Hromkovič J (2004) Algorithmics for hard problems. Springer, Berlin
- Huisman D, Kroon LG et al (2005) Operations research in passenger railway transportation. Stat Neerl 59(4):467–497
- Jansen LN, Pedersen MB, Nielsen OA (2002) Minimizing passenger transfer times in public transport timetables. In: 7th conference of the Hong Kong society for transportation studies: transportation in the information age, pp 229–239
- Kennedy JR (1995) Solving unweighted and weighted bipartite matching problems in theory and practice. In: PhD thesis, Stanford University
- Kohrt JS, Larsen KS (2005) On-line seat reservations via off-line seating arrangements. Int J Found Comput Sci 16(2):381–397
- Kuhn HW (1955) The hungarian method for the assignment problem. Naval Res Log Quart 2:83-97
- Li Z, Zhao J, Peng Q (2021) Train service design in an urban rail transit line incorporating multiple service routes and multiple train compositions. Trans Res Part C 123:102959
- Liu T, Cats O, Gkiotsalitis K (2021) A review of public transport transfer coordination at the tactical planning phase. Trans Res Part C 133:103450
- Miyazaki S, Okamoto K (2010) Improving the competitive ratios of the seat reservation problem. In: Calude CS, Sassone V (eds) Theoretical computer science: 6th IFIP TC 1/WG 2.2 international conference, TCS 2010. Springer, IFIP advances in information and communication technology, pp 328–339
- Neufeld JS, Scheffler M, Tamke F et al (2021) An efficient column generation approach for practical railway crew scheduling with attendance rates. Eur J Oper Res 293(3):1113–1130
- Nieuwenhuis T (2021) Public transport transfer time minimization in luxembourg. In: Bachelor thesis, Vrije Universiteit Amsterdam
- Pyrga E, Schulz F, Wagner D et al (2007) Efficient models for timetable information in public transportation systems. ACM J Exper Algor 12:2.4:1-2.4:39
- Schwartz J, Steger A, Weißl A (2005) Fast algorithms for weighted bipartite matching. In: Nikoletseas SE (ed) Experimental and efficient algorithms. WEA 2005., lecture notes in computer science, vol 3503. Springer, Berlin

- Solanki A, Patidar H (2019) Railway seat allocation system using iterative method. Int J Comput Sci Eng 7(5):815–823
- Zhou Y, Wang Y, Yang H et al (2019) Last train scheduling for maximizing passenger destination reachability in urban rail transit networks. Trans Res Part B Methodol 129:79–95

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