The Complexity of Integrating Routing Decisions in Public Transportation Models

Marie Schmidt and Anita Schöbel

Institute of Numerical and Applied Mathematics
University of Göttingen {m.schmidt,schoebel}@math.uni-goettingen.de

Abstract

To model and solve optimization problems arising in public transportation, data about the passengers is necessary and has to be included in the models in any phase of the planning process. Many approaches assume a two-step procedure: in a first step, the data about the passengers is distributed over the public transportation network using traffic-assignment procedures. In a second step, the actual planning of lines, timetables, etc. takes place. This approach ignores that for most passengers there are many possible ways to reach their destinations in the public transportation network, thus the actual connections the passengers will take depend strongly on the decisions made during the planning phase. In this paper we investigate the influence of integrating the traffic assignment procedure in the optimization process on the complexity of line planning and aperiodic timetabling. In both problems, our objective is to maximize the passengers' benefit, namely to minimize the overall travel time of the passengers in the network. We present new models, analyze NP-hardness results arising from the integration of the routing decisions in the traditional models, and derive polynomial algorithms for special cases.

1998 ACM Subject Classification G.2.2 Network Problems

Keywords and phrases Line Planning, Timetabling, Routing

Digital Object Identifier 10.4230/OASIcs.ATMOS.2010.156

1 Passenger-oriented planning using OD-data

Decisions in public transportation depend strongly on the behavior of the passengers who want to travel in the public transportation network. Thus integrating passenger data in public transportation models in a realistic way is crucial. Until now, many approaches assume a two-step procedure: in a first step, the data about the passengers is distributed over the public transportation network using traffic assignment procedures. In line planning, e.g., one ends up with so called *traffic loads* w_e giving an (approximate) number of passengers who want to use edge e. Also in timetabling it is usually assumed that the number of passengers who want to take a certain vehicle at a certain station is known beforehand. In a second step, the actual planning of lines, timetables, etc. takes place. This reduces the complexity of the models but is not realistic from a practical point of view since the routing decisions of the passengers depend on the lines or timetables which are not known before the optimization problem is solved.

Only a few approaches integrate the routing decisions. In line planning this has been done in [1, 13, 10, 8]. In periodic timetabling this has been studied recently in [4, 6, 3].

In this paper we reformulate some of the common models for line planning and timetabling taking into account origin-destination data and including the routing of the passengers in the optimization process. Thereby we assume that we have passenger data given as a set of *origin-destination-pairs (OD-pairs)* with weights representing the number of passengers traveling from an origin to a destination.

10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS '10).
Editors: Thomas Erlebach, Marco Lübbecke; pp. 156–169
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl Publishing, Germany

The remainder of this paper is structured as follows. We start in Section 2 describing the model we use for the line planning problem integrating the routing decisions, in the following called *line planning with OD-pairs*. As planning with OD-pairs is NP-hard even in special cases, we restrict ourselves to the case of only one OD-pair in Section 2.1 and show its similarity to a Resource-Constrained Shortest Path problem, hence still being a hard problem. In Section 2.2 we furthermore restrict the structure of the public transportation network to be linear. We present polynomial and pseudo-polynomial algorithms for special cases of this problem, and extend them in parts to the case of OD-pairs all having the same origin in Section 2.3.

In Section 3 we introduce a model for aperiodic timetabling, that does not fix passenger weights before the optimization step but integrates the passenger routing into the optimization process. In the following we will call this problem *(aperiodic) timetabling with OD-pairs*. Surprisingly, if the origin events and destination events of the passengers are given the problem turns out to be as easily solvable as the classical timetabling problem, see Section 3.2. However, if origins and destinations of the OD-pairs are given as stations, integrating the passenger routing results again to be strongly NP-hard even if all passengers start at the same station.

2 Line planning with OD-pairs

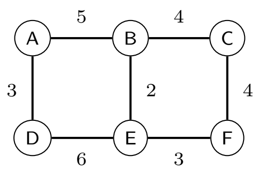
In line planning we consider a *public transportation network* PTN $= (S, E)$ with *stations* $S = \{s_i : i = 1 \ldots, n\}$ and passengers' demand for traveling. The goal of line planning is to determine a set of *lines* $\mathcal{L}'$ and their frequencies. There exist cost-oriented and passenger-oriented objective functions, where the latter may consider the number of direct passengers or the travel time of the passengers. A few recent approaches allow that passengers are freely routed (see [13, 10, 8]).

In our study we investigate the following model. We are given a *line pool* $\mathcal{L}$ from which lines can be chosen. Every line in the pool is given by a directed path in the network that contains every edge at most once. The cost of building a line l is b_l. We also have a set of *OD-pairs* OD $= \{(u_i, v_i) : i = 1, \ldots, m\}$, where $(u, v) \in$ OD represents passengers who want to travel from station $s(u)$ to station $s(v)$. There is a weight w_{uv} assigned to each OD-pair representing the number of passengers who want to travel from $s(u)$ to $s(v)$. For the sake of simplicity we neglect capacity restrictions in this model, and we assume that all chosen lines run with the same frequency. So our objective is

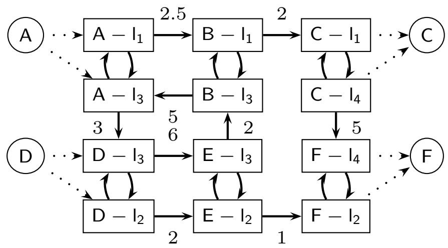
$$\min \sum_{(u,v)\in \mathrm{OD}} w_{uv} W(s(u), s(v)) \text{ s.t. } \sum_{l \in \mathcal{L}'} b_l \leq B$$

where $W(s(u), s(v))$ stands for the *travel time* of OD-pair (u, v). This travel time typically includes the riding time and a penalty for every transfer. Given a length d_{ij} for every edge $\{s_i, s_j\} \in E$ and a *velocity factor* α_l for every line $l \in \mathcal{L}$ the driving time c^l_{ij} of line l on edge $\{i, j\}$ can be determined as $c^l_{ij} := \alpha_l d_{ij}$. The *transfer penalties* $p^{l_1 l_2}_i$ are assumed to depend on the station s_i where the transfer takes places and on the lines l_1 and l_2 between which it is performed. In the constraint we require that the cost of the line system, obtained by summing up the costs b_l for all lines l which are chosen, does not exceed a given *budget* B.

In order to depict the various travel possibilities from the origins to the destinations, we construct a *change&go* network $N = (V, A)$ from the public transportation network PTN (based on [13]). The node set V consists of nodes $[s_i, l]$ for every node $s_i \in S$ and every line l that contains station s_i. For every line l given by the node sequence $s_{1^l}, \ldots, s_{k^l}$ we connect $[s_{j^l}, l]$ to $[s_{j+1^l}, l]$ by a directed arc of length $c_{([s_{j^l},l],[s_{j+1^l},l])} = c^l_{s_{j^l} s_{j+1^l}}$ for every $j = 1, \dots k^l - 1$, representing the driving time for using line l on edge $\{s_{j^l}, s_{j+1^l}\}$. Additionally every pair of nodes $[s_i, l_1]$, $[s_i, l_2]$ is connected by two directed transfer arcs $([s_i, l_1], [s_i, l_2])$ and $([s_i, l_2], [s_i, l_1])$ which represent the transfer possibilities between the lines l_1 and l_2 at station s_i. Thus their arc lengths are $c_{([s_i,l_1],[s_i,l_2])} = p_i^{l_1 l_2}$ and $c_{([s_i,l_2],[s_i,l_1])} = p_i^{l_2 l_1}$, respectively. To model the passengers' demand we add extra nodes u, v for every origin u and every destination v and connect them by directed arcs of travel time and cost 0 to the nodes $[s(u), l]$ and $[s(v), l]$ for all $l \in \mathcal{L}$ respectively. In Figure 1 you can find an example: The public transportation network is depicted in Fig. 1a. The nodes represent stations, the edges represent possible direct rides. We have a line pool $\mathcal{L} = \{l_1 : A - B - C, l_2 : D - E - F, l_3 : A - D - E - B - A, l_4 : C - F\}$ and OD-pairs (A, F) and (D, C). In Fig. 1b the change&go network is shown. The dotted lines represent the origins and destinations of the passengers, the dashed lines stand for the transfer possibilities between two lines.



(a) Public transportation network.



(b) Constructed change&go network.

Figure 1 Construction of the network N from an instance of line planning with OD-pairs.

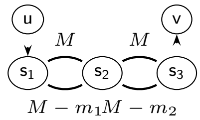
In [13] it has been shown that line planning with OD-pairs is NP-hard even for the case of a linear graph PTN with edge lengths $d_e = 0$ for all $e \in E$, and line costs, transfer penalties and passenger weights all equal to 1. It was also mentioned that line planning stays NP-hard, if all possible lines are included in the line pool.

In order to understand the border between NP-hardness and polynomially computability we will hence make restrictions on the set of OD-pairs. We will start in Section 2.1 with the case where there is only one OD-pair. In Section and 2.2 we further restrict ourselves to the case of linear networks. We will then extend some of the results to the case with several OD-pairs having the same origin and going to the same direction.

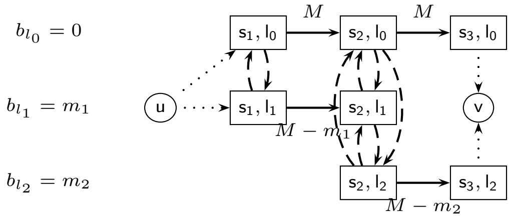
2.1 Line Planning with one OD-pair

For solving the line planning problem with one OD-pair, we assign to each transfer arc $a = ([s_i, l_1], [s_i, l_2])$ and each origin arc $a = (u, [s(u), l_2])$ in the change&go network a second label b_a that represents the line cost $b_a = b_{l_2}$. For the driving arcs and the destination arcs this cost label is set to $b_a = 0$.

Now in this modified change&go network N, we have to find a path from the OD-pair's origin to its destination that satisfies the budget constraint and minimizes the path length which represents the travel time on the path. Thus at first glance our problem looks like a *Resource-Constrained Shortest Path* problem in the change&go network N, where "shortest" is meant with respect to the travel time and the "resource" is the budget B. But still there is a difference: in the line planning problem with one OD-pair, the line cost has only to be paid once, even if a line is entered more often. Nevertheless, we can benefit from known results



(a) Public transportation network.



(b) Change&go network.

Figure 2 Reduction from Partition to line planning with one OD-pair with equal line speed and without transfer penalties.

about the Resource-Constrained Shortest Path problem. Modifying a proof for NP-hardness of this problem (see [14]) shows NP-hardness of line planning with one OD-pair. We then modify a procedure proposed by [9] for the Resource-Constrained Shortest Path problem to solve the line planning problem with one OD-pair in pseudo-polynomial time.

▶ **Theorem 1.** *Line Planning with one OD-pair is NP-hard, even if*
- *the speed of all lines is equal and*
- *there are no transfer penalties.*

Proof. An instance of the decision problem *Partition* [2] consists of a set $\mathcal{M}$ of n numbers that sum up to a number M. The question is whether there is a subset $\mathcal{M}'$ of $\mathcal{M}$ such that the sum of all elements in $\mathcal{M}'$ is $\frac{M}{2}$. Let $\mathcal{M} = \{m_1, \ldots, m_n\}$ be an instance of Partition.

We construct the following instance of line planning with one OD-pair: The public transportation network consists of of $n+1$ stations $s_1, s_2, \ldots, s_{n+1}$. For $j = 1, \ldots n$ s_j is connected to s_{j+1} by two edges, e^j and e_j. The length of e^j is set to M, the length of e_j to $M - m_j$. The line pool consists of $n+1$ lines, $l_0 = (s_1, e^1, s_2, e^2, \ldots, e^n, s_{n+1})$ with cost 0 and $l_j = (s_j, e_j, s_{j+1})$ for $j = 1, \ldots, n$ with cost m_j. Figure 2 shows the PTN and the change & go graph N for an example with $\mathcal{M} = \{m_1, m_2\}$.

Note that for every path from u to v in N we have that the sum of its costs $b(P)$ and its time $c(P)$ is nM. Now we will show that if and only if there is a path with line cost $\leq \frac{M}{2}$ and time $\leq nM - \frac{M}{2}$ there is a solution to the given instance of Partition. Let P be such a path from the origin u to the destination v in the change&go network N, and let E_P be its edge set. From $b(P) + c(P) = nM$, $b(P) \leq \frac{M}{2}$ and $c(P) \leq nM - \frac{M}{2}$ we may conclude that $b(P) = \frac{M}{2}$ holds. Hence, the subset $\mathcal{M}' := \{m_i : i \in I'\}$ of $\mathcal{M}$ is a solution to the given instance of Partition. Vice versa for a solution $\mathcal{M}'$ to Partition we define

$$\mathcal{L}' := \{l_i : m_i \in \mathcal{M}'\} \cup \{l_0\} \quad \text{and} \quad E := \{e_i : m_i \in \mathcal{M}'\} \cup \{e^i : m_i \notin \mathcal{M}'\}.$$

Then E forms a path P in PTN which uses exactly the lines of $\mathcal{L}'$ (since $\mathcal{M}' \neq \mathcal{M}$) and it holds that

$$b(P) = \sum_{l \in \mathcal{L}'} b_l = \sum_{m_i \in \mathcal{M}} m_i = \frac{M}{2}, \quad \text{and} \quad c(P) = \sum_{e \in E} c_e = \sum_{m_i \in \mathcal{M}'} (M - m_i) + \sum_{m_i \notin \mathcal{M}'} M = M - \frac{M}{2},$$

thus opening all lines that are used by P is a solution to the line planning problem. ◀

Although the Resource-Constrained Shortest Path problem is NP-hard, there exist several pseudo-polynomial algorithms (see e.g. [9]). To solve the line planning problem with one OD-pair, we will proceed analogously to [9], that is we will construct a *search graph* in which we are able to run a modified Dijkstra's algorithm in pseudo-polynomial time.

We construct the search graph $G_S^C = (\mathcal{V}_S^C, \mathcal{A}_S^C)$ from N in the following way: For every $i \in V \setminus \{u,v\}$ and every $c \in \{1,2,\ldots,C\}$ with $C = \sum_{a \in A} c_a$ we introduce a node $[i,c]$. For every arc (i_1,i_2) in $A \setminus \{(u,i),(i,v) : i \in V\}$ and every $c \in \{1,2,\ldots,C\}$ with $c + c_{(i_1,i_2)} \leq C$ we draw an arc $([i_1,c],[i_2,c+c_{(i_1,i_2)}])$ and assign a cost of $b_{([i_1,c],[i_2,c+c_{(i_1,i_2)}])} := b_{(i_1,i_2)}$ to it. We introduce a node $[u,0]$ that is connected to all nodes $[i,0]$ for which $(u,i) \in A$ by an arc of length and cost 0. For every $c \in \{1,2,\ldots,C\}$ we add a node $[v,c]$ and connect it to all $[i,c]$ for which $(i,v) \in A$ by an arc of length and cost 0.

We now run a modified Dijkstra's algorithm in this graph to find shortest paths from $[u,0]$ to all nodes $[i,c]$. The traditional Dijkstra's algorithm is modified in the following way: for each node i, for which a path of minimal cost $B(i)$ is already known, a list of the lines that were used to reach this node is stored. Let $\mathcal{P}(k)$ denote the set of nodes for which a minimal cost path is already found in step k. Then for every node j in the set of nodes that are not in $\mathcal{P}(k)$ but adjacent to a node in $\mathcal{P}(k)$, a temporary cost $\tilde{B}(j) = \min_{i \in \mathcal{P}(k)} B(i) + \tilde{b}_{(i,j)}$ is assigned, where

$$\tilde{b}_{(i,j)} = \begin{cases} 0 & \text{if the line associated to } j \text{ is in the list associated to } i \\ b_{(i,j)} & \text{otherwise .} \end{cases}$$

Now, like in the traditional Dijkstra's algorithm, the node j with smallest $\tilde{B}(j)$ is included in $\mathcal{P}(k+1)$ and $B(j) = \tilde{B}(j)$. Among the paths from $[u,0]$ to $[v,c]$ for some c with $B([v,c]) \leq B$ we choose a path $\tilde{P}$ with minimal c and transform it to a path P in N with length a and cost $B([v,a])$ by taking the vertices and arcs corresponding to the ones in $\mathcal{P}$. The result is the calculation of an optimal path in time $O(n_N^2 C^2)$ (as in [9]) where n_N denotes the number of nodes in the network N.

Similarly for $\hat{B} = \min\{B, \sum_{a \in A} b_a\}$ we can construct a modified search graph $G_S^{\hat{B}} = (\mathcal{V}_S^{\hat{B}}, \mathcal{A}_S^{\hat{B}})$ where there is a node for every possible combination of nodes in N and cost values and find a solution using Dijkstra's algorithm in this graph in $O(n_N^2 \hat{B}^2)$.

Thus we obtain the following theorem:

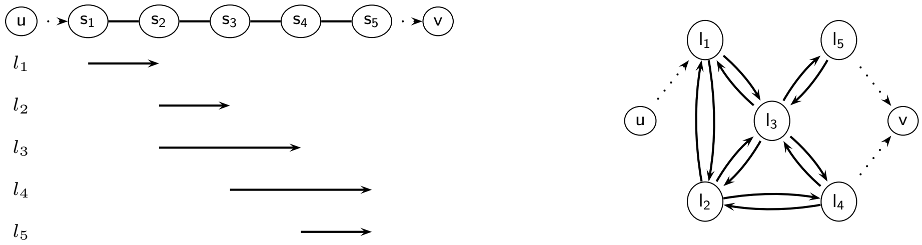
▶ **Theorem 2.** *Let PTN be a public transportation network with n nodes and N the corresponding change&go network with n_N nodes. Then the line planning problem with one OD-pair in N is solvable in pseudo-polynomial time*

1. $O(n_N^2 C^2)$ *with* $C = \sum_{a \in A} c_a$*, or*
2. $O(n_N^2 \hat{B}^2)$ *with* $\hat{B} = \min\{B, \sum_{a \in A} b_a\}$*.*

2.2 Line planning with one OD-pair in a linear network.

In this section we will restrict ourselves to public transportation networks $\text{PTN} = (S,E)$ that are linear, that means $S = \{s_1, s_2, \ldots, s_n\}$ and $E = \{\{s_1,s_2\},\{s_2,s_3\},\ldots,\{s_{n-1},s_n\}\}$. In this case, if all lines have the same speed, it makes no sense for a passenger to leave a line and enter it again later. We hence can apply the solution methods for the Resource-Constrained Shortest Path problem without any modifications.

For linear networks where all lines have the same speed, we can also perform modified Resource-Constrained Path calculations in a the following reduced network $N_{\mathcal{L}}$: Let $S(l) \subset S$ denote the stations that are visited by line l. We define the directed network $N_{\mathcal{L}} = (V_{\mathcal{L}}, A_{\mathcal{L}})$, called *line network*. $V_{\mathcal{L}} = \mathcal{L} \cup \{u,v\}$, that means the nodes of this network are given by the lines of the original problem and the origin and destination node. We set $A_{\mathcal{L}} = \{(l_i,l_j) :$



(a) Linear public transportation network. **(b)** Line network.

Figure 3 Example for the construction of the line network.

$S(l_i) \cap S(l_j) \neq \emptyset\}$ with $b_{(l_i,l_j)} = b_{l_j}$, $b_{(u,l_j)} = b_{l_j}$ and $b_{(l_i,v)} = c_{(u,l_i)} = c_{(l_i,v)} = 0$. See Figure 3 for an example. $N_{\mathcal{L}}$ can be generated from PTN in $O(n|\mathcal{L}|)$.

The arcs in the line network $N_{\mathcal{L}}$ depict the transfer possibilities between the lines, thus we want the length of a path $P_{\mathcal{L}}$ from u to v in $N_{\mathcal{L}}$ to reflect the transfer penalties on this path. As these penalties do not only depend on the lines, but also on the stations where the transfers take place, these penalties are path dependent.

For every path P from u to v in the change&go network N we can find a corresponding path $\boldsymbol{P'_{\mathcal{L}}} = P_{\mathcal{L}}(P)$ from u to v in $N_{\mathcal{L}}$. We define the costs of $P'_{\mathcal{L}}$ to be

$$c(P'_{\mathcal{L}}) = \min\{c(P) : P_{\mathcal{L}}(P) = P'_{\mathcal{L}}\} - \sum_{e \in P_{\mathrm{PTN}}} c_e,$$

for the path P_{PTN} from $s(u)$ to $s(v)$ in PTN. Because of the equal line speed, $\sum_{e \in P_{\mathrm{PTN}}} c_e$ is the driving time for every path P from u to v in N, thus $c(P'_{\mathcal{L}})$ indeed reflects the transfer penalties on a path P in N which is minimal among all paths using the line sequence given by $P'_{\mathcal{L}}$. Note that also for the budget labels b_a it holds that $\sum_{a \in P_{\mathcal{L}}} b_a = \sum_{a \in P(P_{\mathcal{L}})} b_a$.

Thus a path $P_{\mathcal{L}}$ from u to v in $N_{\mathcal{L}}$ of minimal costs $c(P_{\mathcal{L}})$, fulfilling the budget constraints, corresponds to an optimal path P in N with costs $c(P) = c(P_{\mathcal{L}}) + \sum_{e \in P_{\mathrm{PTN}}} c_e$.

This correspondence enables us to improve the run time $O(n_N^2 C^2)$ or $O(n_N^2 \hat{B}^2)$ for the line planning problem with one OD-pair (u, v) in a linear network PTN with all lines having the same speed by solving a modified Resource-Constrained Shortest Path problem in $N_{\mathcal{L}}$:

▶ **Theorem 3.** *A line planning problem with one OD-pair (u, v) in a linear network PTN with equal line speed can be solved in pseudo-polynomial time $O(n|\mathcal{L}| + |\mathcal{L}|^2(Q+1)^2)$ or $O(n|\mathcal{L}| + |\mathcal{L}|^2 \hat{B}^2)$ with $Q = \sum_{s \in S} \sum_{\{l_i,l_j\}, l_i,l_j \in \mathcal{L}(s)} p_s^{l_i l_j}$ and $\hat{B} = \min\{B, \sum_{\{l_i,l_j\} \in A_{\mathcal{L}}} b_{\{l_i,l_j\}}\}$.*

Proof. We construct the line search graph $G_S^Q = (\mathcal{V}_S^Q, \mathcal{A}_S^Q)$ from the line network $N_{\mathcal{L}} = (V_{\mathcal{L}}, A_{\mathcal{L}})$ in the following way: Let $\mathcal{L}(s)$ denote the set of all lines that visit station s. For every $l \in V_{\mathcal{L}} \setminus \{u, v\}$ and every $q \in 1, 2, \ldots, Q$ with $Q = \sum_{s \in S} \sum_{\{l_i,l_j\}, l_i,l_j \in \mathcal{L}(s)} p_s^{l_i l_j}$ we introduce a node $[l, q]$. For every arc (l_1, l_2) in $A_{\mathcal{L}} \setminus \{(u, l), (l, v) : l \in V_{\mathcal{L}}\}$ and every $0 \leq q \leq \tilde{q} \leq Q$ we draw a potential arc from $[l_1, q]$ to $[l_2, \tilde{q}]$. We introduce a node $[u, 0]$ that is connected to all nodes $[l, 0]$ for which $(u, l) \in A_{\mathcal{L}}$ by an arc of length and cost 0. For every $q \in \{1, 2, \ldots, Q\}$ we add a node $[v, q]$ and connect it to all $[l, q]$ for which $(l, v) \in A_{\mathcal{L}}$ by an arc of length and cost 0.

We now run a modified Dijkstra's algorithm in this graph to find shortest paths from $[u, 0]$ to all nodes $[l, q]$. For each node l, for which a path of minimal cost $B(l)$ is already known, a station $s(l)$, representing the current transfer station in the corresponding path P

in N is stored. Let $\mathcal{P}(k)$ denote the set of nodes for which a minimal cost path is already found in step k. Let $\mathcal{T}(k)$ denote the set of nodes $[l_2, \tilde{q}]$ such that $([l_1, q], [l_2, \tilde{q}]) \in \mathcal{A}_S^Q$ for an $[l_1, q] \in \mathcal{P}(k)$ and $\tilde{q} = q + \min_{\hat{s} \geq s(l_1), \hat{s} \in S(l_1) \cap S(l_2)} p_{\hat{s}}^{l_1 l_2}$ with $S(l)$ containing all stations that are visited by line l.That means for given $[l_1, q]$ and l_2, among the potential arcs $([l_1, q], [l_2, \tilde{q}])$ we choose the one that reflects the lowest transfer penalty that is possible for a transfer between l_1 and l_2 after station $s(l_1)$ and include it in $\mathcal{P}(k)$. To every node $[l_2, \tilde{q}]$ in $\mathcal{T}(k)$ a temporary cost $\tilde{B}([l_2, \tilde{q}]) = \min_{[l,q] \in \mathcal{P}(k)} B([l, q]) + b_{l_2}$ is assigned. Now the node $[l_2, \tilde{q}]$ with smallest $\tilde{B}([l_2, \tilde{q}])$ is included in $\mathcal{P}(k+1)$ and $B([l_2, \tilde{q}]) := \tilde{B}([l_2, \tilde{q}])$. Furthermore, we set $s(l_2) := \hat{s}$ for the $\hat{s}$ chosen as a transfer station from l_1 to l_2.Among the paths from $[u, 0]$ to $[v, q]$ for some q with $B([v, q]) \leq B$ we choose a path P_S with minimal q and transfer it to a path $P_{\mathcal{L}}$ in $N_{\mathcal{L}}$ with length q and cost $B([v, q])$.

Like in the original Dijkstra's algorithm we have to consider every arc in the line search graph at most once, so the run time is quadratic in the number of nodes of G_S^Q. Similarly we can construct a modified line search graph $G_S^{\hat{B}} = (\mathcal{V}_{\mathcal{L}}^{\hat{B}}, \mathcal{A}_{\mathcal{L}}^{\hat{B}})$ where there is a node for every possible combination of node in N and cost value and find a solution using Dijkstra's algorithm in this graph in $O(|\mathcal{L}|^2 \hat{B}^2)$. ◀

Note that if the transfer penalties do not depend on the stations where the transfers take place, they can be assigned directly as lengths to the arcs of the line network such that the problem can be solved directly as a Resource-Constrained Shortest Path problem.

But still, line planning in a linear network with one OD-pair is NP-hard.

▶ **Theorem 4.** *Line Planning with one OD-pair in a linear public transportation network is NP-hard, even if the speed of all lines is equal.*

▶ **Theorem 5.** *Line Planning with one OD-pair in a linear public transportation network is NP-hard, even if there are no transfer penalties.*

For the proofs of these two results we refer to [12].

Combining the two restrictions from Theorems 4 and 5, due to Theorem 3 we however obtain a polynomial run time of $O(n|\mathcal{L}| + |\mathcal{L}|^2)$ in the case without transfer penalties and $O(n|\mathcal{L}| + |\mathcal{L}|^6)$ with equal penalties which can further be improved as follows.

▶ **Lemma 6.** *Line planning with one OD-pair in a linear public transportation network with equal line speed can be solved in*

1. $O(n|\mathcal{L}| + |\mathcal{L}|^2)$ *if there are no transfer penalties.*
2. $O(n|\mathcal{L}| + |\mathcal{L}|^4)$ *if the transfer penalties are all equal.*

Proof. The first statement follows directly from Theorem 3 or by applying the Dijkstra's algorithm in $N_{\mathcal{L}}$ to find a cost optimal solution.

For the second statement, without loss of generality we can assume $c_{\text{change}} = 1$. $N_{\mathcal{L}}$ has $|\mathcal{L}| + 2$ nodes. As a shortest path in $N_{\mathcal{L}}$ visits every node at most once, the optimal travel time in $N_{\mathcal{L}}$ is bounded by $Q = |\mathcal{L}|$. So according to Theorem 3 given the line network $N_{\mathcal{L}}$ the problem can be solved in $O(|\mathcal{L}|^4)$. ◀

2.3 Line planning with OD-pairs having the same origin in linear networks

In this section we investigate if the results of the previous section can be generalized. We still stick to the restriction that the underlying network is linear but relax the strong assumption of only one OD-pair by allowing a set of OD-pairs which all have the same origin and start traveling into the same direction. We will see that in some cases we still can apply the algorithms for one OD-pair from Sections 2.1 and 2.2 so that we can solve some problems easily. However, having several OD-pairs with the same origin, line planning is even NP-hard if all lines have the same speed and if all transfer penalties are equal (which for one OD-pair can be solved in polynomial time, see Lemma 6).

▶ **Theorem 7.** *Line Planning with OD-pairs having the same origin and going to the same direction in a linear public transportation network is NP-hard, even if*
- *the speed of all lines is equal and*
- *all transfer penalties are equal.*

Proof. The proof is a reduction from Partition similar to Theorem 2, see [12] for details. ◀

In the following lemma we will show that in the situation of Theorem 7, there is an optimal solution such that the paths of all OD-pairs are nested. This property will enable us to solve the problem analogously to a line planning problem with only one OD-pair with equal line speed in a linear network in pseudo-polynomial time.

▶ **Lemma 8.** *Consider a line planning problem with all OD-pairs having the same origin and going to the same direction in a linear public transportation network with*
- *equal line speed and*
- *equal transfer penalties.*

There is always an optimal line set $\mathcal{L}'$ together with a set of paths $\{P_i^*\}$ in $N(\mathcal{L}')$, P_i^* being the path for OD-pair (u, v_i) without origin and destination arc, such that $P^* := \bigcup_{(u,v_i)\in OD} P_i^*$ is a path in N.

Proof. Because of the equal line speed, the driving time for the OD-pairs is not path dependent. Thus instead of the total travel time, we can regard only the weighted sum of the transfers.

Suppose that $\{P_i : i = 1, \ldots, m\}$ is the path set of an optimal solution where P_i is the path from $s(u)$ to $s(v_i)$. Now (assuming that the OD-pairs are ordered such that the distance from $s(u)$ to $s(v_i)$ increases with increasing i) we will show that for every P_m that is contained in such a set, there exist paths P_i^* for $i = 1, \ldots, m-1$ such that $P_i^* \subset P_m$ and $\{P_i^* : i = 1, \ldots, m-1\} \cup \{P_m\}$ is also an optimal path set for the problem.

Suppose that this is not the case. Then in every optimal path set there exists an index i such that $P_i \not\subset P_m$. Let P_i^m be the path from $s(u)$ to $s(v_i)$ contained in P_m. For a subgraph G of N we denote by $b(G)$ the sum of the costs of all lines used by G and by $t(G)$ the number of transfers in G. Concerning the line costs we first observe that $b(P_i \cup P_m) \geq b(P_m) = b(P_m \cup P_i^m)$. Thus $t(P_i^m) > t(P_i)$, because otherwise changing P_i to P_i^m would lead to an optimal set. As $p_s^{l_i l_j} := p$ for all transfers we have $t(P_i) + p \leq t(P_i^m)$.If we denote by $\hat{P}_m$ the path consisting of P_i, possibly a transfer and the part $P_m(v_i)$ of P_m starting in station $s(v_i)$, we obtain

$$t(\hat{P}_m) \leq t(P_i) + p + t(P_m(v_i)) = t(P_i) + p + t(P_m) - t(P_i^m) \leq t(P_m)$$

and thus for the transfer costs weighted with the passenger numbers it holds that

$$w_i t(P_i) + w_m t(\hat{P}_m) < w_i t(P_i) + w_m t(P_m).$$

Thus the total transfer costs for path set $\{P_j : j = 1, \ldots, m-1\} \cup \{\hat{P}_m\}$ are smaller than the total transfer costs for path set $\{P_i : i = 1, \ldots, m\}$. From $b(P_i^m \cup P_m) \geq b(P_i^m \cup \hat{P}_m)$ it follows that

$$b(\{P_i : i = 1, \ldots, m\}) \geq b(\{P_j : j = 1, \ldots, m-1\} \cup \{\hat{P}_m\}),$$

thus the path set $\{P_j : j = 1, \dots, m-1\} \cup \{\hat{P}_m\}$ is feasible. Thus $\{P_i : i = 1, \dots, m\}$ was not an optimal path set. ◀

This property enables us to find an optimal solution reducing the problem to a line planning problem with one OD-pair and applying Theorem 3.

▶ **Theorem 9.** *The line planning problem with all OD-pairs having the same origin and going to the same direction in a linear public transportation network with*

- *equal line speed for all lines and*
- *equal transfer penalties*

is solvable in pseudo-polynomial time $O(n|\mathcal{L}| + |\mathcal{L}|^2\hat{B}^2)$ *or* $O(n|\mathcal{L}| + |\mathcal{L}|^2W^2)$ *where*

$$\hat{B} = \min\{B, \sum_{a\in A} b_a\}, \quad \textit{and} \quad W = \sum_{s\in S} \sum_{\{l_i,l_j\},l_i,l_j\in\mathcal{L}(s)} \sum_{(u,v_j)\in OD, s<s(v_j)} w_{uv_j}.$$

Proof. Consider an instance I_1 of the described problem with OD-pairs $(u, v_i) \in \mathrm{OD}_1$ labeled in increasing order of $s(v_i)$. Let N_1 denote the associated change&go network. Let I_2 denote the instance of the line planning problem in the same public transportation network with the same costs, with the only OD-pair $(u, v_m) \in \mathrm{OD}_1$ for which the distance between $s(u)$ and $s(v_j)$ is maximal and where the transfer penalties are given as

$$p^{l_il_j}_{s_k} := p_{s_k} := \sum_{j=1,\dots,m:(u,v_j)\in OD:s_k<s(v_j)} w_{uv_j}.$$

Let N_2 denote the associated change&go network. We will now show that there is a bijection between solution paths P^2 in N_2 and sets of solution paths $\{P^1_i : i = 1, \dots, m\}$ in N_1 with $P^1 := \bigcup_{(u,v_i)\in\mathrm{OD}} P^1_i$ being a path in N_1, both having the same line costs and the same solution value. Let $b(G)$ denote the line costs of a subgraph G of a change&go network and $t(P')$ the number of transfers on a path P'. For a path P^2 in N_2 define $P^1_i(P^2)$ to consist of the path P^2 seen as path in N_1 ending as soon as the station $s(v_i)$ is reached. We directly obtain $b(P^2) = b(\bigcup_{i=1}^m P^1_i(P^2))$. Furthermore it can be justified that $t(P^2) = \sum_{j=1}^m w_{uv_j} t(P^1_j(P^2))$.Vice versa, for a set of paths $\{P^1_i : i = 1, \dots, m\}$ in N_1 for which $P^1 := \bigcup_{(u,v_i)\in\mathrm{OD}} P^1_i$ is a path, we define $P^2(P^1)$ as the path P^1 regarded in N_1. Then like above we have

$$b(P^2(P^1)) = b(P^1) = b(\bigcup_{i=1}^m P^1_i) \quad \text{and} \quad t(P^2(P^1)) = \sum_{j=1}^m w_{uv_j} t(P^1_j).$$

Thus there is a bijection between solution paths P^2 in N_2 and sets of solution paths $\{P^1_i : i = 1, \dots, m\}$ in N_1 with $P^1 := \bigcup_{(u,v_i)\in\mathrm{OD}} P^1_i$ being a path in N_2, both having the same line costs and the same solution value. Finally, Theorem 9 follows by applying Theorem 3 to the instance I_2 of the line planning problem for one OD-pair and all lines having the same speed that is constructed in the way described above. ◀

In Lemma 6.1 it has been shown that the line planning problem with one OD-pair in a linear public transportation network can be solved by a Dijkstra's algorithm regarding the line costs if all lines have the same speed and there are no transfer penalties, because in this case the value of the objective function does not depend on the choice of $\mathcal{L}'$. For the case of multiple OD-pairs with the same origin, we obtain a minimal cost solution by applying the algorithm from Lemma 6.1 for the OD-pair with longest travel time.

▶ **Lemma 10.** *A line planning problem with all OD-pairs having the same origin and going to the same direction in a linear public transportation network*

- *with equal line speed and*
- *without transfer penalties*

can be solved in $O(n|\mathcal{L}| + |\mathcal{L}|^2)$.

Similar to these results we can also use Theorem 2 to derive a pseudo-polynomial algorithm for the case of arbitrary line speed and no transfer penalties.

Our results of line planning in linear networks are summarized in the following table.

Restrictions line speed	Restrictions transfer penalties	Complexity one OD-pair	Complexity same origin
equal	no penalties	$O(n\|\mathcal{L}\| + \|\mathcal{L}\|^2)$ (6.1)	$O(n\|\mathcal{L}\| + \|\mathcal{L}\|^2)$ (10)
equal	equal penalties	$O(n\|\mathcal{L}\| + \|\mathcal{L}\|^4)$ (6.2)	NP-hard (7), solvable in $O(n\|\mathcal{L}\| + \|\mathcal{L}\|^2 W^2)$ or $O(n\|\mathcal{L}\| + \|\mathcal{L}\|^2 \hat{B}^2)$ (9)
equal	*arbitrary*	NP-hard (4), solvable in $O(n\|\mathcal{L}\| + \|\mathcal{L}\|^2 (Q+1)^2)$ or $O(n\|\mathcal{L}\| + \|\mathcal{L}\|^2 \hat{B}^2)$ (3)	NP-hard (4, 7)
arbitrary	no penalties	NP-hard (5), solvable in $O(n_N^2 C^2)$ or $O(n_N^2 \hat{B}^2)$ (2)	NP-hard (5), solvable in $O(n_N^2 \hat{B}^2)$ or $O(n_N^2 \tilde{C}^2)$ [12]
arbitrary	equal penalties or *arbitrary*	NP-hard (5), solvable in $O(n_N^2 C^2)$ or $O(n_N^2 \hat{B}^2)$ (2)	NP-hard (5)

where

- $C = \sum_{a \in A} c_a$,
- $\tilde{C} := \sum_{i=1}^{n-1} (\sum_{j=1,\dots,m:(u,v_j)\in OD: s_{i+1} < s(v_j)} w_{uv_j}) \cdot c_{[s_i,l],[s_{i+1},l]}$,
- $\hat{B} = \min\{B, \sum_{a\in A} b_a\}$,
- $Q = \sum_{s\in S} \sum_{\{l_i,l_j\}, l_i,l_j \in \mathcal{L}(s)} p_s^{l_i l_j}$, and
- $W = \sum_{s\in S} \sum_{\{l_i,l_j\}, l_i,l_j\in\mathcal{L}(s)} \sum_{(u,v_j)\in \mathrm{OD}, s<s(v_j)} w_{uv_j}$.

Note that the case of equal line speed and without transfer penalties can still be solved in polynomial time for general OD-pairs, see [12].

3 (Aperiodic) Timetabling with OD-pairs

Given a line plan, the timetabling process searches for the arrival and departure times for all lines at all stations. To this end the public transportation network PTN is extended to a so-called *event-activity-network* $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ (see e.g. [7, 5]). Every arrival and every departure of a vehicle is modeled as an *arrival* or *departure event* $e \in \mathcal{E} = \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}$. The events are connected by *driving activities* $\mathcal{A}_{\text{drive}}$, *waiting activities* $\mathcal{A}_{\text{wait}}$, or *changing activities* $\mathcal{A}_{\text{change}}$. If π_i denotes the time of event i, and $a = (i,j)$ an activity linking event i and event j, a *timetable* $(\pi_i)_{i\in\mathcal{E}}$ is feasible if every activity $a = (i,j)$ satisfies $l_a \le \pi_j - \pi_i \le u_a$ for some given lower and upper bounds l_a and u_a on the duration of activity a. While in periodic timetabling (see [5] and references therein) it is required that the resulting timetable is periodic which causes NP-hardness even for the feasibility problem, aperiodic timetabling drops the assumption of feasibility which in general results in an easier problem. Given a fixed number of passengers w_a for every activity a, the goal of traditional timetabling is to minimize the sum of the travel times. The problem can be solved efficiently by linear programming [11]. In our model we again do not start with such fixed weights w_a but with a set of OD-pairs which can be freely routed through the network. For integer programming as well as heuristic approaches for periodic timetabling with OD-pairs see [4, 6, 3].

3.1 Integrating OD-pairs in the model

Let a set of OD-pairs $\mathrm{OD} = \{(u, v)\}$ with weights w_{uv} be given. Every path from a departure event i at station u to a departure event j at station v represents a possible journey from u to v. As in general $\pi_i \neq \pi_{i'}$ and $\pi_j \neq \pi_{j'}$ for departure events i and i' at station u and arrival events j and j' at station v, the travel time depends on the path that is chosen for the OD-pair (u, v). To integrate the routing procedure we add the origins and destinations to the network by introducing *origin nodes* $\mathcal{E}_{\text{org}} = \{u^{\text{org}} : (u, v) \in \mathrm{OD}\}$ and *destination nodes* $\mathcal{E}_{\text{dest}} = \{v^{\text{dest}} : (u, v) \in \mathrm{OD}\}$. We connect every $u^{\text{org}} \in \mathcal{E}_{\text{org}}$ to all departure events i at station u by an *origin arc* (u^{org}, i) and every arrival event j at station v to $v^{\text{dest}} \in \mathcal{E}_{\text{dest}}$ by a *destination arc* (j, v^{dest}). The arc sets are denoted by $\mathcal{A}_{\text{org}}$ and $\mathcal{A}_{\text{dest}}$ respectively.

Our objective is to find a feasible timetable π and for every OD-pair (u, v) a path $P_{uv} = (u^{\text{org}}, i_1^{uv}, i_2^{uv}, \ldots, i_{uv}^{uv}, v^{\text{dest}})$ from u^{org} to v^{dest} such that the sum of all travel times $\sum_{(u,v)\in\mathrm{OD}} w_{uv}(\pi_{i_{uv}^{uv}} - \pi_{i_1^{uv}})$ is minimal.

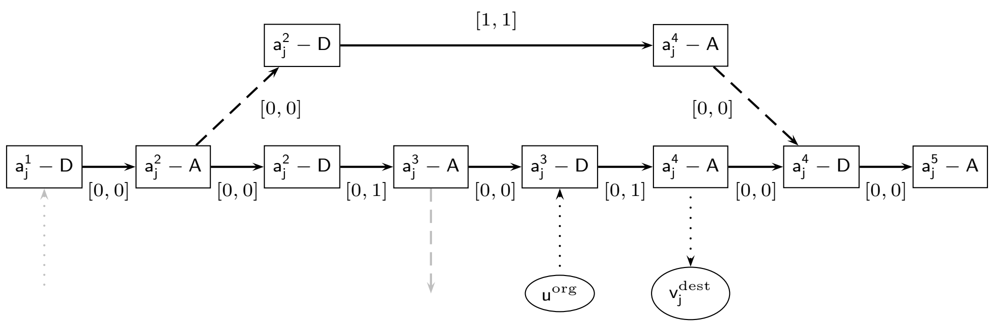
Finding an optimal solution to the described problem turns out to be strongly NP-hard:

▶ **Theorem 11.** *The timetabling problem with OD-pairs is strongly NP-hard, even if all OD-pairs have the same origin.*

Proof. An instance of the decision problem *Minimum Cover* ([2]) consists of a finite set S, a collection C of subsets of S and a positive integer $K \leq |C|$. The question to decide is whether there is a subset C' of C with $|C'| \leq K$ such that every element of S is contained in at least one member of C'.

Let $m = |S|$ and $n = |C|$. To reduce an instance (S, C, K) of Minimum Cover to the timetabling problem with OD-pairs for every $s_i \in S$ we will represent the elements $s_i \in S$ by OD-pairs (u, v^i) with $w_{uv^i} = n$ and the sets $c_j \in C$ by a structure consisting of two trains tr_j^1 and tr_j^2, five stations $a_j^1, a_j^2, a_j^3, a_j^4, a_j^5$ and an OD-pair (u, v_j) with $w_{uv_j} = 1$ in the way depicted in Figure 4. Here, the square nodes are the departure and arrival events. The origin and destination events are represented by ovals. The dotted lines are the origin and destination arcs, the solid lines represent driving and waiting activities, changing activities are represented by dashed lines. The gray lines indicate where it will be possible to enter and to leave this structure.

Note that for each of these structures str_i when making the timetable we have to choose either to assign a length of 1 to the arc $(a_j^2 - D, a_j^3 - A)$ or to the arc $(a_j^3 - D, a_j^4 - A)$. If we assign a length of 1 to the latter, the travel time of OD-pair (u, v_j) will be 1 and because of $w_{uv_j} = 1$ it will contribute an amount of 1 to the objective function.



■ **Figure 4** The structure str_j representing a set c_j in the reduction from Minimum Cover to the timetabling problem with OD-pairs.

We identify a_j^1 and u^{org} for all stations a_j^1, that means we connect every a_j^1 by an origin activity to the origin node u^{org}. For every $s_i \in c_j$ we connect $(a_j^3 - A)$ to a departure event $(a_j^3 - Dep)$ of a train tr_{ij} that by a driving activity is connected to the arrival event of train tr_{ij} in v^i. The upper and lower bound for all arcs outside of the structure str_j are set to $[l_a, u_a] = [0, 0]$. See Figure 4 and 5 for an example of the construction for an instance of Minimum Cover with $S = \{1, 2, 3, 4\}$ and $C = \{\{2, 3, 4\}, \{1, 4\}, \{2, 3\}\}$. The square nodes are the departure and arrival events. The origin and destination events are represented by ovals. The dotted lines are the origin and destination arcs, the solid lines represent driving and waiting activities, changing activities are represented by dashed lines. The nodes $struc_j$ represent the structures from Figure 4.

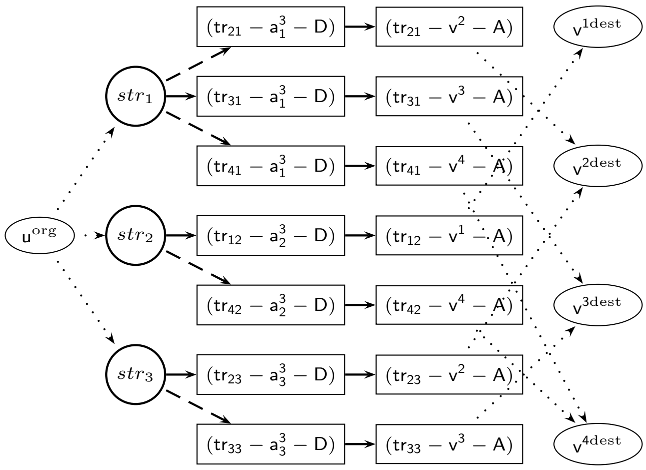


Figure 5 Reduction from Minimum Cover.

We observe that if for an OD-pair (u, v_i) there is a structure str_j such that u^{org} is connected to str_j and there is a length of 1 assigned to $(a_j^3 - D, a_j^4 - A)$, the OD-pair will arrive at its destination in time 0 while if there is no such structure, there will be a contribution of n to the objective function. Thus as $K < |C| = n$ in a feasible solution for every OD-pair there must be at least one structure str_j such that u is connected to str_j and there is a length of 1 assigned to $(a_j^3 - D, a_j^4 - A)$.

We conclude that $C' := \{c_{j_1}, \ldots, c_{j_k}\}$ is a solution to the Minimum Cover problem if and only if assigning a length of 1 to $(a_j^3 - D, a_j^4 - A)$ for all j such that $c_j \in C'$ and to $(a_j^2 - D, a_j^3 - A)$ for all other j leads to a solution of the timetabling problem with OD-pairs with solution value $\leq K$. ◀

3.2 Timetabling with routing between events

Let's assume that instead of having a set of OD-pairs consisting of pairs of stations we have a set of OD-pairs that consists of departure and arrival events as origins and destinations. I.e., the passengers not only fix the location of their origins and destinations but also the departure and arrival events (the first and the last train they wish to take). Thus OD $= \{(i, j)\}$, where $i \in \mathcal{E}_{\text{dep}}$ represents the departure of the train a passenger wants to take at a certain station and $j \in \mathcal{E}_{\text{arr}}$ represents the arrival of the train at the end of the passenger's journey. Again we assign a weight w_{ij} to (i, j). Note that there may be several paths in $\mathcal{N}$ connecting a departure event i to an arrival event j, thus given OD-pair (i, j) we do not know the specific path this OD-pair will take. Nevertheless, the choice between these paths does not really matter because in the resulting timetable these paths will all have the same length of $\pi_j - \pi_i$. Our objective is to minimize the weighted sum of the travel times of all OD-pairs:

$$\min \sum_{(i,j)\in \mathrm{OD}} w_{ij} \cdot (\pi_j - \pi_i) \tag{1}$$

$$s.t. \quad \pi_h - \pi_g \in [l_{gh}, u_{gh}] \qquad \forall (g,h) \in \mathcal{A} \tag{2}$$

$$\pi_g \in \mathbb{Z} \qquad \forall g \in \mathcal{E} \tag{3}$$

▶ **Theorem 12.** *Aperiodic timetabling with OD-pairs given as origin and destination events can be solved by linear programming.*

Proof. The coefficient matrix of the problem is the transposed of a node-arc-incidence matrix and hence totally unimodular. ◀

We can envision the minimization of the weighted sum of the $\pi_j - \pi_i$ in terms of the original problem by introducing virtual edges from i to j for every OD-pair (i, j) and assigning weights w_{ij} to these edges and $w_a = 0$ to all other edges. Formulating the original aperiodic timetabling problem in this modified network leads to the formulation (1)-(3).

Note that the travel time of an OD-pair (i, j) only depends on the time at node i and node j and not on the path from i to j. If for every OD-pair (i, j) we chose a path P_{ij} from i to j, set $w_a := \sum_{(i,j)\in \mathrm{OD}: a\in P_{ij}} w_{ij}$, solving the aperiodic timetabling problem with weights w_a leads again to the same integer program since the objective functions are equal.

We can use the result from Theorem 12 to solve the general timetabling problem with OD-pairs: Let $\mathcal{N}$ be a network and $\mathrm{OD} = \{(u_i, v_i) : i = 1, \ldots, n\}$ be a set of OD-pairs. In the network $\mathcal{N}$ for every OD-pair (u_i, v_i) we define $\mathcal{E}^i_{\mathrm{dep}} := \{e \in \mathcal{E}_{\mathrm{dep}} : (u_i^{\mathrm{org}}, e) \in \mathcal{A}_{\mathrm{org}}\}$ and $\mathcal{E}^i_{\mathrm{arr}} := \{e \in \mathcal{E}_{\mathrm{arr}} : (e, v_i^{\mathrm{dest}}) \in \mathcal{A}_{\mathrm{dest}}\}$.

▶ **Lemma 13.** *The timetabling problem with OD-pairs can be solved by solving every different instance $(\mathcal{N}, \tilde{OD})$ of the timetabling problem with OD-pairs $OD := \{(e^i_{dep}, e^i_{arr}) : i = 1, \ldots, n\}$ with $e^i_{dep} \in \mathcal{E}^i_{dep}, e^i_{arr} \in \mathcal{E}^i_{arr}$ and comparing the solution values. In particular*

1. *If for every $(u_i, v_i) \in OD$ it holds that $|\mathcal{E}^i_{dep}| = |\mathcal{E}^i_{arr}| = 1$ an optimal solution to the timetabling problem with OD-pairs can be found by solving one linear program.*
2. *If $OD = \{(u_1, v_1)\}$ an optimal solution to the timetabling problem with OD-pairs can be found by solving at most $|\mathcal{E}^1_{dep}| \cdot |\mathcal{E}^1_{arr}|$ linear programs.*

4 Conclusions and Further Research

In this paper we integrated the routing of the passengers in the optimization process in line planning and timetabling problems. We showed that solving the line planning problem with OD-pairs is NP-hard even in simplified cases, but were able to give polynomial time algorithms for several special cases. Although timetabling with fixed passenger paths can be solved easily by linear programming, including the routing decisions results in an NP-hard problem. However, if the start and destination event of every OD-pair are known, the problem can be solved as efficiently as aperiodic timetabling itself.

In our further research, we will generalize the algorithms for line planning with one OD-pair in linear networks to develop heuristics for the general case. An important next step will be to include capacity restrictions and frequencies in the line planning model. Concerning timetabling, we are interested in finding further restrictions on the problem structure that make the problem easily solvable and to develop heuristics based on these approaches. Another promising heuristic idea, both for line planning and timetabling, is to iterate routing and optimization steps to successively improve the solution; we are currently testing such a procedure numerically. We furthermore will investigate the benefit of such an integrating, e.g., the improvement of the passengers' travel time when integrating routing in the optimization process.

References

1 J.A. Mesa G. Laporte, A. Marín and F.A. Ortega. An integrated methodology for rapid transit network design. In *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, pages 187–199. Springer, 2007.

2 M.R. Garey and D.S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

3 M. Kaspi. Service oriented train timetabling. Master's thesis, Tel Aviv University, 2010.

4 M. Kinder. Models for periodic timetabling. Master's thesis, Technische Universität Berlin, 2008.

5 C. Liebchen. *Periodic Timetable Optimization in Public Transport*. PhD thesis, Technische Universität Berlin, 2006. published by dissertation.de.

6 J. Lübbe. Passagierrouting und taktfahrplanoptimierung. Master's thesis, Technische Universität Berlin, 2009. in german.

7 K. Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables*. Deutsches Zentrum für Luft– und Raumfahrt, Institut für Flugführung, Braunschweig, 1998. Habilitationsschrift.

8 K. Nachtigall and K. Jerosch. Simultaneous network line planning and traffic assignment. In M. Fischetti and P. Widmayer, editors, *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, 2008.

9 C. A. Phillips. The network inhibition problem. In *Annual ACM Symposium on Theory of Computing - Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 776–785, New York, 1993. ACM.

10 M. E. Pfetsch R. Borndörfer, M. Grötschel. A column generation approach to line planning in public transport. *Transportation Science 41*, pages 123–132, 2007.

11 R. T. Rockafellar. *Network flows and monotropic optimization*. John Wiley & Sons, Inc., 1984.

12 M. Schmidt and A. Schöbel. The complexity of integrating routing decisions in public transportation models. Technical report, Institut für Numerische und Angewandte Mathematik, Georg-August Universität Göttingen, 2010. NAM Report.

13 A. Schöbel and S. Scholl. Line planning with minimal transfers. In *5th Workshop on Algorithmic Methods and Models for Optimization of Railways*, number 06901 in Dagstuhl Seminar Proceedings, 2006.

14 Z. Wang and J. Crowcroft. Quality of service routing for supporting multimedia applications. *IEEE Journal on Selected areas in communications*, 14(7):1228–1234, 1996.