Line Planning with Minimal Transfers

Anita Schöbel
Susanne Scholl
Institut für Numerische und Angewandte Mathematik
Georg-August Universität Göttingen

November 12, 2004

Abstract

An important strategic element in the planning process of public transportation is the development of a line concept, i.e. to find a set of paths for operating lines on them. So far, the models in the literature aim to minimize the costs or to maximize the number of direct travelers. In this paper we present a new approach minimizing the travel times over all customers including penalties for the transfers needed. This approach maximizes the comfort of the passengers and will make the resulting timetable more reliable. To tackle our problem we present integer programming models and suggest a solution approach using Dantzig-Wolfe decomposition for solving the LP-relaxation. Numerical results of real-world instances are presented.

1 Motivation and related literature

In the strategic planning process of a public transportation company one important step is to find a suitable line concept, i.e. to define the routes of the bus or railway lines. Given a public transportation network PTN $= (S, E)$ with its set of stations S and its set of direct connections E, a *line* is defined as a path in this network. The *line concept* is the set of all lines offered by the public transportation company, together with their *frequencies*, where the frequency f_l of a line l contains the number of vehicles serving line l within the planning period considered. The *frequency of an edge* e, on the other hand, is the number of vehicles running along the edge.

The line planning problem has been well studied in the literature. For an early contribution we refer to Dienst, see [Die78]. The many models given after this time can be roughly classified into the following two types. In a *cost-oriented approach* the goal is to find a line concept serving all customers and minimizing the costs for the public transportation company. The basic cost model has been suggested in Claessens et al., see [CvDZ96], where a binary (linear) programming formulation has been given. A solution approach by branch and cut has been developed in [GvHK01]. In [Bus97] an alternative formulation with integer variables has been proposed. In [BLL04] Bussieck et al. present a fast solution approach combining nonlinear techniques with integer programming. More recently, [GvHK02] takes also into account different types of vehicles simultaneously. A new approach is to take into account that the behavior of the customers depends on the design of the lines. A first cost-oriented model including such demand changes was treated with simulated annealing in two diploma theses in cooperation with *Deutsche Bahn*, see [Kli00, Sch01].

On the other hand, in the *direct travelers approach* by Bussieck et al. [BKZ96] (see also [Bus97]) the goal is to maximize the number of direct travellers (i.e. customers that need not change the line to reach their destination). As constraint, the number of vehicles running along an edge is restricted for each edge in the PTN, i.e. upper and lower bounds on the allowed frequencies on each edge are taken into account.

Although the latter model is a customer-oriented approach it maximizes the amount of one group of customers but without considering the remaining ones which might have very many transfers during their trips. It also does not take into account the travel times for the customers: Sometimes it is preferable to have a transfer but reach the destination earlier instead of sitting in the same line for the whole trip but having a large detour.

A recent work by Quak [Qua03] treats line planning for buses instead of trains. He develops a two phase algorithm with the construction of the lines in the first and setting of frequencies and departure times in the second phase. In contrary to the other models he is not taking lines out of a given line pool but constructs them from the scratch, which is the main part of his work. The two main objectives of this model are "minimizing the total drive time of the vehicles" to keep the costs for the company low and "minimizing the mean detour time of the passenger requests" to keep the passengers comfort high since short travel times are requested by the passengers. As he sets up also a timetable in the second phase, he tries to keep the changing times low to couple the second objective. But if the changing times are low, the risk of loosing a connection in case of a small delay in the network is very high. Also we have to mention that a transfer is a bigger discomfort than a slightly longer travel time.

In this work we take into account these points. Based on our presentations ([SS03] and [SS04]) we develop a new model which allows to sum over all travel times over all customers including penalties for the transfers needed. We also show how different frequencies of the lines can be taken into account. The remainder of the paper is organized as follows. In Section 2 we introduce the new line planning model, discuss its complexity in Section 3 and then describe an integer programming model in Section 4.

Our solution approach which is based on a Dantzig-Wolfe decomposition is given in Section 5. Finally, we present numerical results based on a real-world application of German Rail (DB).

2 Basic definitions

A *public transportation network* is a finite, undirected graph $\mathrm{PTN} = (S, E)$ with a node set S representing stops or stations, and an edge set E, where each edge $\{u, v\}$ indicates that there exists a direct ride from station u to station v (i.e., a ride that does not pass any other station in between). For each edge $\{u, v\}$ we assume that the driving time t_{uv} is known.

We assume the PTN as already given and fixed. We further assume that a *line pool* $\mathcal{L}$ is given, consisting of a set of paths in the PTN. Each line $l \in \mathcal{L}$ is specified by a sequence of stations, or, equivalently, by a sequence of edges. Let $E(l)$ be the set of edges belonging to line l. Given a station $u \in S$ we furthermore define

$$\mathcal{L}(u) = \{l \in \mathcal{L} : u \in l\}$$

as the set of all lines passing through u.

Moreover, let $\mathcal{R} \subseteq S \times S$ denote the set of all origin-destination pairs (s, t) where w_{st} is the number of customers wishing to travel from station s to station t.

The line planning problem then is to choose a subset of lines $L \in \mathcal{L}$, together with their frequencies, which

- allows each customer to travel from its origin to its destination,
- is not too costly, and
- minimizes the "inconvenience" for the customers.

In the literature, the only customer-oriented approach dealing with the inconvenience of the customers is the approach of [Bus97] (see also [BKZ96]) in which the number of direct travellers is maximized. In our paper, however, we deal with the sum of *all* transfers over all customers. On a first glance, the problem to minimize the number of transfers seems to be similar to maximizing the number of direct travellers. That is in general not the case, as the following example demonstrates.

Example 2.1. Given a PTN with 9 nodes and 8 edges as shown in Figure 1 and a line pool $\mathcal{L}$ containing 11 lines $\mathcal{L} = \{l_1, \ldots, l_{11}\}$ shown in Table 1. In Figure 1 for simplicity only lines l_1, l_2 and l_3 are named. The remaining lines correspond to the single edges. Let the set of origin-destination pairs be $\mathcal{R} := \{(1, 3), (2, 8), (7, 9)\}$ with weights $w_{1,3} = w_{2,8} = w_{7,9} = 1$. Assume that due to safety rules not more than one vehicle per edge is allowed within our planning period of e.g. 30 minutes.
Then the optimal solutions for the two objectives are the following line concepts:

line	stations
l_1	1,2,3
l_2	7,8,9
l_3	2,3,4,5,6,7,8
l_4	1,2
l_5	2,3
l_6	3,4
l_7	4,5
l_8	5,6
l_9	6,7
l_{10}	7,8
l_{11}	8,9

Table 1: The line pool of example 2.1.

- "maximize number of direct travellers": $L = \{l_2, l_3, l_6, \ldots, l_9\}$
  In this case the two passengers $(1,3)$ and $(7,9)$ can travel directly, but passenger $(2,8)$ has to change 5 times.
- "minimize number of transfers": $L = \{l_1, l_4, l_{11}\}$
  In this case only one passenger, namely passenger $(2,8)$ travels directly, but the total number of transfers is only two because passengers $(1,3)$ and $(7,9)$ have to change once each.

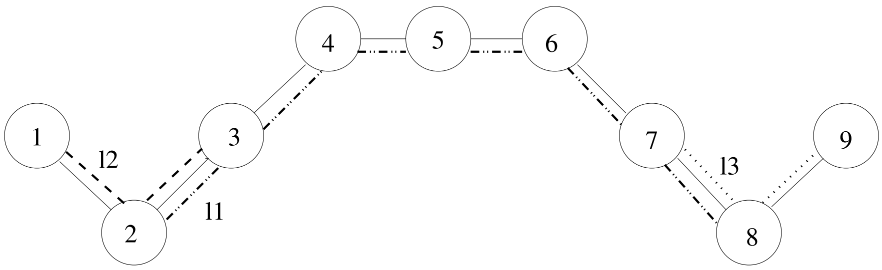

Figure 1: Difference between the objectives "maximize direct travellers" and "minimize transfers".

Note that considering the number of transfers only would lead to solutions with very long lines, serving all origin-destination pairs directly but having large detours for the customers. To avoid this we determine not only a line concept, but also a path for each origin-destination pair and count the number of transfers **and** the length of the paths in the objective function. This is specified next.

Given a set of lines $L \subseteq \mathcal{L}$, a customer can travel from its origin s to its destination t, if there exists an s-t-path P in the PTN only using edges in

$\{E(l) : l \in L\}$. The "inconvenience" of such a path is then approximated by the weighted sum of the traveling time along the path and the number of transfers, i.e.

$$\text{inconvenience}(P) = k_1 \text{Time}_P + k_2 \text{Transfers}_P.$$

On the other hand, the cost of the line concept $L \subseteq \mathcal{L}$ is calculated by adding the costs C_l for each line $l \in L$, assuming that such costs C_l are known beforehand.

The *line planning problem* hence is to find a feasible set of lines $L \subseteq \mathcal{L}$ together with a path P for each origin-destination pair, such that the costs of the line concept do not exceed a given budget B and such that the sum of all inconveniences over all paths is minimized.

Since the capacity of a vehicle is not arbitrarily large, we have to extend the basic problem to include frequencies of the lines. This makes sure that there are enough vehicles along each edge to transport all passengers. If each origin-destination pair can be served, the line concept is called *feasible.* We remark that often, the number of vehicles running along the same edge is also bounded from above, e.g., for safety reasons.

3 Complexity Results

In this section we first show that the line planning problem as defined above is NP-hard, even in a very simple case, corresponding to $k_1 = 0$ in the above definition.

Theorem 3.1. *The line planning problem is NP-complete, even if*

- *we only count the number of transfers in the objective function,*
- *the* PTN *is a linear graph.*
- *all costs C_l are equal to one.*

Proof. In the decision version, the line planning problem in the above case can be written as follows:

Given a graph PTN $=(S, E)$ with weights c_e for each $e \in E$, origin-destination pairs $\mathcal{R}$, and a budget B, does there exist a feasible set of B lines with less than K transfers?

We reduce the set covering problem to the line planning problem: Given a set covering problem in its integer programming formulation

$$\min\{\underline{1}x : Ax \geq \underline{1}, x \in \{0,1\}^n\}$$

with an 0-1 $m \times n$ matrix A, we construct a line planning problem as follows:

We define the PTN as a linear graph with $2m$ nodes $S = \{s_1, t_1, s_2, t_2 \ldots, s_m, t_m\}$ and edges $E = \{(s_1, t_1), (t_1, s_2), (s_2, t_2), (t_2, s_3), \ldots, (s_m, t_m)\}$. We define an origin-destination pair for each row of A,

$$\mathcal{R} = \{(s_i, t_i) : i = 1, \ldots, m\}.$$

For column j of A we construct a line l_j passing through nodes s_i and t_i if $a_{ij} = 1$.

As an example, Figure 2 shows the line planning problem obtained from a set covering problem with

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

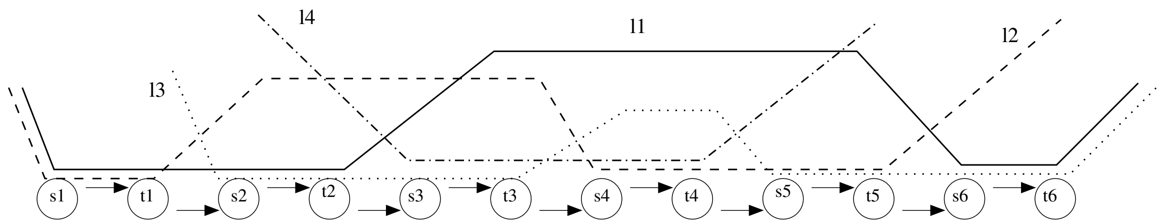

Figure 2: Construction of the line planning problem in the proof of Theorem 3.1.

Setting $K = 0$ we hence have to show that a cover with less than B elements exists if and only if the line planning problem has a solution in which all passengers can travel without changing lines. Due to our construction this is true and hence the theorem holds. ∎

A question that might arise in this context, is what happens if the lines need not be chosen from a given line pool, but can be constructed as any path. Some of the basic cost models become very easy in this case, but unfortunately, the complexity status of the line planning problem treated in this paper does not change.

Theorem 3.2. *The line planning problem in which* **all** *possible simple paths are allowed is NP-complete, even if we only count the number of transfers in the objective function.*

Proof. We reduce the Hamiltonian path problem to our problem.

Let $G = (\mathcal{V}, \mathcal{E})$ be the graph in which we want to check the existence of a Hamiltonian path from a given node s to a given node t. We construct the line planning problem as follows:

We define the PTN as the given graph G and construct

$$\mathcal{R} = \{(s,v),(v,t) : v \in \mathcal{V} \setminus \{s,t\}\}$$

as the set of origin-destination pairs. Furthermore, we set the budget $B = 1$. The line planning problem with $K = 0$ hence is to find **one** line serving all origin-destination pairs. Such a line must start in s, pass through all nodes and end in t (otherwise at least one element of $\mathcal{R}$ would have to change for its trip), and hence constitutes a Hamiltonian path. Vice versa, any Hamiltonian path is a solution of the line planning problem with a total of zero transfers. ∎

4 A model for the line planning problem

To model the line planning problem as integer program we use the PTN to construct a directed graph, the so-called *change&go network* $G_{CG} = (\mathcal{V}, \mathcal{E})$ as follows:

We extend the set S of stations to a set $\mathcal{V}$ of nodes with nodes representing either station-line-pairs (change&go nodes: $\mathcal{V}_{CG}$) or the start and end points of the customers (origin-destination nodes: $\mathcal{V}_{OD}$), i.e. $\mathcal{V} := \mathcal{V}_{CG} \cup \mathcal{V}_{OD}$ with

- $\mathcal{V}_{CG} := \{(s,l) \in S \times \mathcal{L} : l \in \mathcal{L}(s)\}$ (set of all station-line-pairs)
- $\mathcal{V}_{OD} := \{(s,0) : (s,t) \in \mathcal{R} \text{ or } (t,s) \in \mathcal{R}\}$ (origin-destination nodes)

The new set of edges $\mathcal{E}$ consists of directed edges between nodes of the same stations (representing that customers board or unboard a vehicle or change lines) and edges between nodes of the same line (representing the driving activities):

$$\mathcal{E} := \mathcal{E}_{change} \cup \mathcal{E}_{OD} \cup \mathcal{E}_{go}$$

with

- $\mathcal{E}_{change} := \{((s,l_1),(s,l_2)) \in \mathcal{V}_{CG} \times \mathcal{V}_{CG}\}$ (changing edges)
- $\mathcal{E}_l := \{((s,l),(s',l) \in \mathcal{V}_{CG} \times \mathcal{V}_{CG} : (s,s') \in E\}$ (driving edges of line $l \in \mathcal{L}$)
- $\mathcal{E}_{go} := \bigcup_{l \in \mathcal{L}} \mathcal{E}_l$ (driving edges)
- $\mathcal{E}_{OD} := \{((s,0),(s,l)) \in \mathcal{V}_{OD} \times \mathcal{V}_{CG} \text{ and } ((t,l),(t,0)) \in \mathcal{V}_{CG} \times \mathcal{V}_{OD} : (s,t) \in \mathcal{R}\}$ (origin-destination edges)

We define weights on all edges $e \in \mathcal{E}$ of the change&go network representing the inconvenience customers have when using edge e. Given a set of lines $L \subseteq \mathcal{L}$ we then can determine the lines the customers are likely to use by calculating a shortest path in the change&go network for each single origin-destination pair. Therefore the choice of the edge costs c_e is very important. Some examples:

1. Customers only count transfers:

$$c_e = \begin{cases} 1 & : \quad e \in \mathcal{E}_{change} \\ 0 & : \quad \text{else} \end{cases}$$

   Note that in this case, it is possible to shrink the change&go network to a network with $|\mathcal{L}| + |S|$ nodes and $|\mathcal{E}_{change}| + |\mathcal{E}_{OD}|$ edges.

2. Real travel time:

$$c_e = \begin{cases} 0 & : \quad e \in \mathcal{E}_{OD} \\ \text{travel time in minutes} & : \quad e \in \mathcal{E}_{go} \\ \text{time needed for changing platform} & : \quad e \in \mathcal{E}_{change} \end{cases}$$

   More specific, to model the line planning problem as defined in Section 2, we set

$$c_e = \begin{cases} 0 & \text{if } e \in \mathcal{E}_{OD} \\ k_1 t_{uv} & \text{if } e = ((u,l),(v,l)) \in \mathcal{E}_{go} \\ k_2 & \text{if } e \in \mathcal{E}_{change} \end{cases}$$

3. Many other extensions are possible.

Since we assume that customers behave selfish we need an implicit calculation of shortest paths (with respect to the weights c_e) within our model. This is obtained by solving the following network flow problem for each origin-destination pair $(s,t) \in \mathcal{R}$.

$$\theta x_{st} = b_{st},$$

where

- $\theta \in \mathbb{Z}^{|\mathcal{V}| \times |\mathcal{E}|}$ is the node-arc-incidence matrix of G_{CG},
- $b_{st} \in \mathbb{Z}^{|\mathcal{V}|}$ is defined by

$$b^i_{st} = \begin{cases} 1 & : \quad i = (s,0) \\ -1 & : \quad i = (t,0) \\ 0 & : \quad \text{else} \end{cases}$$

- and $x^e_{st} \in \{0,1\}$ are the variables, where $x^e_{st} = 1$ if and only if edge e is used on a shortest dipath from node $(s,0)$ to $(t,0)$ in G_{CG}.

To specify the lines in the line concept we introduce variables $y_l \in \{0,1\}$ for each line $l \in \mathcal{L}$, which are set to 1 if and only if line l is chosen to be in the line concept. Our model, *Line Planning with Minimal Travel Times* (LPMT) can now be presented.

(LPMT)

$$\min \sum_{(s,t) \in \mathcal{R}} \sum_{e \in \mathcal{E}} w_{st}\ c_e\ x^e_{st}$$

$$\begin{aligned}
s.t. \sum_{(s,t)\in\mathcal{R}} \sum_{e\in\mathcal{E}_l} x^e_{st} &\le y_l M_l && \forall\ l \in \mathcal{L} && (1)\\
\theta x_{st} &= b_{st} && \forall\ (s,t) \in \mathcal{R} && (2)\\
\sum_{l\in\mathcal{L}} C_l y_l &\le B && && (3)\\
x^e_{st}, y_l &\in \{0,1\} && \forall\ (s,t) \in \mathcal{R}, e \in \mathcal{E}, l \in \mathcal{L} && (4)
\end{aligned}$$

Constraint (1) makes sure that a line must be included in the line concept if the line is used by some origin-dastination pair, where M_l is a sufficiently large number, at least bigger than the number of edges of line l times the number of origin-destination pairs: $M_l \ge |\mathcal{E}_l| \cdot |\mathcal{R}|$. Constraint (2) models the selfish behavior of the customers, i.e., that customers use shortest paths according to the weights c_e.

Having only constraints (1) and (2), the best line concept from a customer-oriented point of view would be to introduce all lines of the line pool. This is certainly no option for a public transportation company, since running a line is costly. Let C_l be an estimation of the costs for running line l and let B be the budget the public transportation company is willing to spend. Then the budget constraint (3) takes the economic aspects into account.

The objective function we use is customer-oriented: We sum up the costs $\sum_{e\in\mathcal{E}} w_{st}\ c_e\ x^e_{st}$ of a shortest path from s to t for each origin-destination pair $(s,t) \in \mathcal{R}$, i.e., we minimize the average costs of the customers.

In (LPMT) we implicitly assume that all customers traveling from station s to station t choose the same path in the change&go network, i.e., the same set of lines. This can be done if edge capacities are neglected in (LPMT). In practice, this is usually not the case, since each vehicle only can transport a limited number of customers and usually there is only a limited number of vehicles possible along each line (e.g. due to safety rules). In the following, we therefore present an extension of (LPMT) taking into account the number of vehicles on each line in a given time period. Consequently, this formulation allows to split customers along different paths from s to t in the change&go network G_{CG}.

Let N denote the capacity of a vehicle and let the new variables $f_l \in \mathbb{N}$ contain the frequency of line l, i.e., the number of vehicles running along line l within a given time period. Furthermore we choose variables $x^e_{st} \in \mathbb{N}$ and change the vector b_{st} to

$$b^i_{st} = \begin{cases} w_{st} & \text{if } i = (s,0)\\ -w_{st} & \text{if } i = (t,0)\\ 0 & \text{else} \end{cases}$$

Then the *Line Planning Model with minimal transfers and frequencies* (LPMTF) is the following:

(LPMTF)

$$\min \sum_{(s,t)\in\mathcal{R}} \sum_{e\in\mathcal{E}} c_e \; x_{st}^e$$

$$
\begin{aligned}
s.t.\quad & \frac{1}{N}\sum_{(s,t)\in\mathcal{R}} x_{st}^e \le f_l && \forall\; l\in\mathcal{L}, e\in\mathcal{E}_l && (1)\\
& \theta x_{st} = b_{st} && \forall\; (s,t)\in\mathcal{R} && (2)\\
& \sum_{l\in\mathcal{L}} C_l f_l \le B && && (3)\\
& \sum_{l\in\mathcal{L}:k\in\mathcal{E}_l} f_l \le f_k^{max} && \forall\; k\in E && (4)\\
& x_{st}^e, f_l \in \mathbb{N} && \forall\; (s,t)\in\mathcal{R}, e\in\mathcal{E}, l\in\mathcal{L} && (5)
\end{aligned}
$$

Constraints (1) make sure that the frequency of a line is high enough to transport the passengers. If $f_l = 0$, the line l is not chosen in the line concept. Constraints (2) are flow conservation constraints routing the passengers on the shortest possible paths. Note that the x_{st}^e variables can take integer values, such that passengers may choose different paths for the same origin-destination pair. Constraint (3) is again the budget constraint but with costs for each vehicle of a line (which are multiplied by the frequency to get the costs of the line).The capacity constraint (4) may be included if upper bounds for the frequencies are present.

5 Dantzig-Wolfe Decomposition

The line planning problem introduced in Section 4 is NP-hard, and, moreover in real-world instances, gets huge (see Section 6). But fortunately (LPMT) as well as (LPMTF) have a block structure with only a few coupling constraints. Moreover, in both models, all blocks (except the one containing the single budget constraint) are totally unimodular since they are network flow problems. We take advantage of this structure by using a Dantzig-Wolfe decomposition. In this section we present our approach for (LPMT). The method can also be applied for solving (LPMTF) since the model structure is very similar. However, the numerical results deal with (LPMT).

We now present the formulation of the master LP and of the corresponding subproblems. For further details on the algorithm the reader is referred to Dantzig and Wolfe [DW60].

The block structure of the model is shown in the following reformulation.

$\min \sum_{(s,t)\in\mathcal{R}} \sum_{e\in\mathcal{E}} w_{st} c_e x_{st}^e$

$$\begin{array}{lllll}
\boxed{\sum_{(s,t)\in\mathcal{R}} \sum_{e\in l} x_{st}^{e} \leq y_l M} & \text{coupling constraints} & & & \\
\quad \boxed{X^{s_1,t_1}} & & & & \\
 & \boxed{X^{s_2,t_2}} & & & \\
 & & \ddots & & \\
 & & & \boxed{X^{s_r,t_r}} & \\
 & & & & \boxed{Y}
\end{array}$$

with $|\mathcal{R}|+1$ blocks:
$X^{st} := \{x_{st} \in \mathbb{Z}^{|\mathcal{E}|} : \theta x_{st} = b_{st}, 0 \leq x_{st}^{e} \leq 1, \ \forall\ e \in \mathcal{E}\}$

$Y := \{y \in \mathbb{Z}^{|\mathcal{L}|} : C^T y \leq B, 0 \leq y_l \leq 1, \ \forall\ l \in \mathcal{L}\}$
and $|\mathcal{L}|$ coupling constraints:

$$\sum_{(s,t)\in\mathcal{R}} A_X \ x_{st} - A_Y \ y \leq 0.$$

The coefficient matrix $(A_X|\ldots|A_X|-A_Y)$ of the coupling constraints looks as follows:

- A_X is an $|\mathcal{L}| \times |\mathcal{E}|$ matrix given by elements $a_{le} = 1$, if $e \in \mathcal{E}_l$, zero otherwise. It is equal for each OD-pair
- A_Y is an $|\mathcal{L}| \times |\mathcal{L}|$ diagonal matrix containing M_l as its lth diagonal element.

With the weight-cost-constants $c_{st}^{e} := w_{st} c_e$ and the $|\mathcal{L}|$-vector v as slack variable we get the following master LP:

(Master 1)

$$z = \min \sum_{(s,t)\in\mathcal{R}} \sum_{i} (c_{st}\ x_{st}^{(i)}) \alpha_{st}^{i}$$

$$\begin{array}{llrr}
s.t. & \sum_{(s,t)\in\mathcal{R}} \sum_i (A_X\ x_{st}^{(i)}) \alpha_{st}^{i} - \sum_i (A_Y\ y^{(i)}) \beta^i + Iv = 0 & & (1) \\
 & \sum_i \alpha_{st}^{i} = 1 & \forall\ (s,t) \in \mathcal{R} & (2) \\
 & \sum_i \beta^i = 1 & & (3) \\
 & v_l, \alpha_{st}^{i}, \beta^i \geq 0 & &
\end{array}$$

where $x_{st}^{(i)}$ and $y^{(i)}$ are the extreme points of X^{st}, and of Y, respectively. This problem has $|\mathcal{L}|$ coupling constraints and $|\mathcal{R}|+1$ convexity constraints.
For each $(s,t) \in \mathcal{R}$ we obtain the following subproblem

$$z_{st} = \min (c_{st} - \pi A_X) x_{st} - \mu_{st}$$

$$\text{s.t.} \quad x_{st} \in X^{st}$$

and the subproblem of the Y-block is

$$z = \min(-\pi A_Y)y - \mu_{00}$$

$$\text{s.t.} \quad y_l \in Y,$$

where $\{\pi_i\}_{i \in \mathcal{L}}$ are the dual variables of the coupling constraints, $\{\mu_{st}\}_{(s,t)\in\mathcal{R}}$ are the dual variables of the alpha convexity constraints and μ_{00} is the dual variable of the beta convexity constraint.

It is also possible to add the budget constraint to the set of coupling constraints and keep the shortest-path constraints in one block. Note that computing the subproblem in this second formulation is as easy as before since it decomposes into $|\mathcal{R}|$ independent smaller problems. However, the number of constraints in the master problem changes to $|\mathcal{L}| + 1$ coupling constraints but only one convexity constraint.

(Master 2)

$$z = \min \sum_i (c_{st}\ x^{(i)})\alpha^i$$

$$\begin{array}{lll} s.t. & \sum_{(s,t)\in\mathcal{R}} \sum_i (A_X\ x^{(i)})\alpha^i - A_Y y_l + Iv = 0 & (1) \\ & \sum_i \alpha^i = 1 & (2) \\ & v_l, \alpha^i, y_l \geq 0 & \end{array}$$

The subproblem is

$$z = \sum_{(s,t)\in\mathcal{R}} \min(c_{st} - \pi A_X)x_{st} - \mu$$

$$\text{s.t.} \quad x_{st} \in X^{st} \quad \forall\ (s,t) \in \mathcal{R}$$

Note that in this case the x^i are not $\{0,1\}$ any more since we sum up over all shortest path solutions in the subproblem.

Since the original and the master formulation may differ in their linear programming relaxations, we will now discuss the strength of the linear programming master (Master 2). The bounds provided by the LP relaxation of the original program (P) and the Master (M) are respectively

$$z_{LP}(P) = \min\{cx : Ax \geq b, Dx \geq d, x \geq 0\}$$

and

$$z_{LP}(M) = \min\{cx : Ax \geq b, x \in conv(\{x \in \mathbb{N}^n : Dx \geq d\})\}.$$

From integer programming theory (see e.g. [Wol98]), we know that

$$z_{LP}(P) \leq z_{LP}(M) \leq z_{IP}$$

where z_{IP} is the integer solution value ($z_{IP} = z_{IP}(P) = z_{IP}(M)$).

When $conv(\{x \in \mathbb{N}^n : Dx \geq d\}) = \{x \in \mathbb{R}^n_+ : Dx \geq d\}$ (i.e. when the current formulation of the subsystem already has the integrality property), $z_{LP}(P) = z_{LP}(M)$.

As we have mentioned, the shortest path blocks X^{st} in (LPMT) as well as the corresponding network flow blocks in (LPMTF) are totally unimodular and thus have the integrality property. We hence get

$$z_{LP}(LPMT) = z_{LP}(\text{Master 2})$$

The same result holds for (LPMTF) with the corresponding (Master 2). As we will see in Section 6 the LP relaxation is not solvable for real world instances due to the size of the resulting change&go network. Therefore a decomposition makes sense even if the provided bound is not better. In our case the LP relaxation is only solvable for small instances, see Table 2 in Section 6.

6 Real-world application and computational results

Our approach is currently tested on instances of the long distance trains of the German railway network. The line pool we use was generated by German railway (DB). The given PTN consists of a line pool of 423 lines, 35322 origin-destination pairs, 233 stations and 319 edges.

This leads to a change&go network with 6705 nodes, 343271 edges and a model with $2.42 \cdot 10^{10}$ variables and 236834434 constraints. We implemented the Dantzig-Wolfe decomposition of our model using Xpress MP 2004 and Microsoft Visual C++ 6.0. The CPU times are based on a 3.06 GB Intel4 processor with 512 MB RAM.

Our computational experience shows that the second variation of the Dantzig-Wolfe decomposition (solving the subproblem with Dijkstra's shortest path algorithm) finds an optimal solution of the LP relaxation within minutes. Even for the largest instances a solution was found in less than two hours. Note that line planning is part of the strategic planning and so even longer computation times of some hours are acceptable. In general only few iterations are needed.

6.1 Solving the LP relaxation

The size of (LPMT) depends on two criteria:

$\|\mathcal{L}\|$	$\|\mathcal{R}\|$=2	$\|\mathcal{R}\|$=10	$\|\mathcal{R}\|$=50	$\|\mathcal{R}\|$=100	$\|\mathcal{R}\|$=150	$\|\mathcal{R}\|$=200	$\|\mathcal{R}\|$=1476
10	0.187	0.860	4.219	8.563	12.469	16.453	154.281
50	4.875	23.610	118.844	239.782	361.063	M	M
100	25.172	124.359	626.640	M	M	M	M
132	52.047	257.906	M	M	M	M	M
150	M	M	M	M	M	M	M

Table 2: CPU times for the LP relaxation of (LPMT) for different line pool sizes and origin-destination pairs.

1. the size of the line pool: the change&go network is constructed out of the line pool and thus the size of the node-arc incidence matrix θ increases with a bigger line pool
2. the number of origin-destination pairs: we solve a shortest path problem for each origin-destination pair and so the number of constraints increases if we treat more origin-destination pairs

In Section 5 we mentioned that the bound provided by the decomposition using (Master 2) is equal to the LP relaxation of (LPMT). But as the LP relaxation of (LPMT) is only solvable for small instances, a decomposition still makes sense. Table 2 shows the CPU times for the LP relaxation of (LPMT) for instances with different sizes of line pools $|\mathcal{L}|$ and origin-destination pairs $|\mathcal{R}|$, using the software Xpress MP. If the problem was not solvable due to lack of memory, this is indicated by "M"[1].

6.2 Variations of the Decomposition

We implemented four different possibilities of a Dantzig-Wolfe decomposition:

1. The budget constraint does not belong to the coupling constraints and we treat the X^{st}-blocks as independent blocks (previously called (Master 1)).
2. The budget constraint belongs to the coupling constraints and we treat the X^{st}-blocks as independent blocks.
3. The budget constraint does not belong to the coupling constraints and we treat the X^{st}-blocks as one big block.
4. The budget constraint belongs to the coupling constraints and we treat the X^{st}-blocks as one big block (previously called (Master 2)).

Our computational experience shows that it is better for solving the complete problem if the budget constraint belongs to the coupling constraints

[1] For Table 2 we used a computer with 2 GB RAM instead of 512MB which was used for all other computations.

Decomposition	No. 0	No. 1	No. 13
LP relax	0.01	M	M
DW 1	0.19	M	M
DW 2	0.15	M	M
DW 3	0.12	4	17318
DW 4	0.1	1	8715

Table 3: CPU times for different decompositions explained in Section 6.2. "M" indicates that the memory was not sufficient.

since it is only one constraint and its variables do not appear in the objective function. Solving the corresponding subproblem takes unnecessary computation time.

Normally it is known to be better to split the remaining constraints into as many blocks as possible in order to reduce the number of iterations and thus the computation time. Of course, the number of convexity constraints in the master problem increases but since the structure of the convexity constraints is very simple, modern optimization software can manage them easily. In our case, the number of convexity constraints increases enourmeously if we treat the X^{st}-blocks individually and so memory problems arise even for small instances. On the other hand, we figured out that even if we treat the shortest path problems as one subproblem and thus only have one convexity constraint, we only need very few iterations such that there is no need for further decomposition.

Table 3 shows computation times of the different decompositions and of the LP-relaxation of (LPMT) for three different instances 0, 1, and 13 explained in Table 4.

6.3 Variations of the line pool

The CPU time depends mainly on the size of the line pool and on the number of origin-destination pairs. We calculated 14 instances of different sizes to show the dependency between the running time and the line pool size. We always used the same origin-destination matrix, but to avoid infeasibility we deleted such origin-destination pairs that could not be served even if all lines of the line pool were used. The remaining number of elements of $\mathcal{R}$ is shown in Table 5 in column "$|\mathcal{R}|$".
In column "CPU", we show the CPU time of an implementation of the 4th variation of the master problem (i.e., the budget constraint belongs to the coupling constraints and the X^{st}-blocks are treated as one big block, previously called (Master 2)), in which we used Dijkstra's algorithm for solving the subproblem. Since we only need very few iterations, we can save a lot of running time by checking whether the π change in comparison to the last iteration or not after solving the subproblems. If not, there

No.	$\|\mathcal{L}\|$	$\|\mathcal{R}\|$	Budget	Stations	Nodes (ext.)	Edges (ext.)
0	3	2	2	3	10	32
1	10	2602	8	220	419	1212
2	50	4766	45	220	1015	11152
3	100	11219	80	242	1716	32080
4	132	18238	100	319	2487	48788
5	200	10126	150	305	3590	138616
6	250	13246	200	307	4716	223034
7	275	14071	250	309	5303	268696
8	300	17507	250	316	5931	316742
9	330	18433	300	318	6706	382906
10	350	17095	300	317	6503	467500
11	375	18350	300	320	7101	544828
12	400	22191	300	325	7682	600348
13	423	22756	400	325	8268	679382

Table 4: Instances for different line pool sizes.

is no need to recalculate the shortest paths in the next iteration since no distances will change. In this case, we just have to adjust the objective value by $-\mu$. The CPU times of this variation are shown in Table 5 in column "CPU check".

6.4 Preprocessing

As we have seen, the main problem of our approach is the size of the change&go network depending mainly on the size of the line pool. A wise choice of a possibly small line pool is therefore advisable. On the other hand it makes sense to analyze the underlying PTN. For example if two lines go parallel for a long time, it is sufficient to add changing edges only at the first and the last station. Also arcs between stations without changing possibility can be shrinked to decrease the size of the network.

7 Conclusions

We developed an integer programming model for the line planning problem that minimizes the travel times over all customers including penalties for the tranfers needed and proposed an extension that includes frequencies. We showed that the problem is NP-hard even if lines are constructed and not chosen from a given line pool. Since the problem gets so huge that a straightforward solution of the LP relaxation is not possible we developed a solution approach based on Dantzig-Wolfe decomposition. Computational

No.	$\vert\mathcal{L}\vert$	$\vert\mathcal{R}\vert$	CPU	CPU check
0	3	2	0.1	0.05
1	10	2602	2	1
2	50	4766	10	3
3	100	11219	52	16
4	132	18238	92	48
5	200	10126	155	78
6	250	13246	689	329
7	275	14071	1193	691
8	300	17507	2167	1171
9	330	18433	3246	1911
10	350	17095	2769	1814
11	375	18350	5150	2727
12	400	22191	6525	4789
13	423	22756	15243	8715

Table 5: CPU times for different line pool sizes.

results for various real world instances and different decompositions were presented. We are currently working on a branch&price algorithm and heuristics to get an integer solution.

References

[BKZ96] M.R. Bussieck, P. Kreuzer, and U.T. Zimmermann. Optimal lines for railway systems. *European Journal of Operational Research*, 96(1):54–63, 1996.

[BLL04] M.R. Bussieck, T. Lindner, and M.E. Luebbecke. A fast algorithm for near cost optimal line plans. *Math. Methods Oper. Res.*, 59(3), 2004.

[Bus97] M.R. Bussieck. *Optimal lines in public transport*. PhD thesis, Technische Universität Braunschweig, 1997.

[CvDZ96] M.T. Claessens, N.M. van Dijk, and P.J. Zwaneveld. Cost optimal allocation of rail passenger lines. *European Journal on Operational Research*, 110, 1996.

[Die78] H. Dienst. *Linienplanung im spurgeführten Personenverkehr mit Hilfe eines heuristischen Verfahrens*. PhD thesis, Technische Universität Braunschweig, 1978.

[DW60] G.P. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[GvHK01] J. Goossens, C.P.M. van Hoesel, and L.G. Kroon. A branch and cut approach for solving line planning problems. Technical Report RM/01/016, University of Maastricht, 2001. METEOR Research Memorandum.

[GvHK02] J. Goossens, C.P.M. van Hoesel, and L.G. Kroon. On solving multi-type line planning problems. Technical Report RM/02/009, University of Maastricht, 2002. METEOR Research Memorandum.

[Kli00] H. Klingele. Verfahren zur Optimierung eines Linienkonzeptes der Deutschen Bahn AG. Master's thesis, Universität Karlsruhe, 2000.

[Qua03] C.B. Quak. Bus line planning. Master's thesis, TU Delft, 2003.

[Sch01] M. Schmidt. Modelle zur Linienoptimierung im Zugverkehr unter Berücksichtigung der Nachfrage. Master's thesis, Universität Kaiserslautern, 2001.

[SS03] A. Schöbel and S. Scholl. Planung von Linien mit minimalen Umsteigevorgängen. In D. Mattfeld, editor, *Proceedings of the GOR-workshop on "Optimierung im öffentlichen Nahverkehr"*, pages 69–89, 2003.

[SS04] A. Schöbel and S. Scholl. Planning lines with minimal transfers - a Dantzig-Wolfe approach. In *Proceedings of TRISTAN V - The Fifth Triennial Symposium on Transportation Analysis, Le Gosier, Guadeloupe, 13-18 June*, 2004.

[Wol98] L.A. Wolsey. *Integer Programming.* John Wiley & Sons, 1998.