Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

RALF BORNDÖRFER   MARTIN GRÖTSCHEL   MARC E. PFETSCH

A Path-Based Model for Line Planning in Public Transport

Supported by the DFG Research Center MATHEON "Mathematics for key technologies"

ZIB-Report 05-18 (March 2005)

A Path-Based Model for Line Planning in Public Transport*

Ralf Borndörfer Martin Grötschel Marc E. Pfetsch

Abstract

The *line planning problem* is one of the fundamental problems in strategic planning of public and rail transport. It consists in finding lines and corresponding frequencies in a transport network such that a given travel demand can be satisfied. There are (at least) two objectives. The transport company wishes to minimize operating costs, the passengers want to minimize travel times. We propose a new multi-commodity flow model for line planning. Its main features, in comparison to existing models, are that the passenger paths can be freely routed and that the lines are generated dynamically. We discuss properties of this model and investigate its complexity. Results with data for the city of Potsdam, Germany, are reported.

1 Introduction

The *strategic planning* process in public and rail transport is usually divided into consecutive steps of *network design*, *line planning*, and *timetabling*. Operations research methods can support the planning decisions in each of these steps, see for instance the survey articles of Odoni, Rousseau, and Wilson [18] and of Bussieck, Winter, and Zimmermann [6].

This article is about the *line planning problem* (LPP) in public transport. The problem is to design line routes and their frequencies in a given street or track network such that a given transportation volume, given by a so-called origin-destination matrix (OD-matrix), can be satisfied. The frequency of a line is supposed to indicate a basic timetable period and controls the lines' transportation capacity. There are two competing objectives: on the one hand to minimize the operating costs of lines and on the other hand to minimize user discomfort. User discomfort is usually measured by the total passenger traveling time or the number of transfers during the ride, or both.

*Supported by the DFG Research Center MATHEON "Mathematics for key technologies"
Address of the authors: Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustr. 7, 14195 Berlin, Germany; Email: `{borndoerfer, groetschel, pfetsch}@zib.de`

The recent literature on the LPP mainly deals with railway networks. One common assumption is the so-called *system split*, which fixes the traveling paths of the passengers *before* the lines are known. A second common assumption is that an optimal line plan can be chosen from a (small) precomputed set of lines. Third, maximization of *direct travelers*, i.e., travelers without transfers, is sometimes considered as the objective. In such an approach, transfer waiting times do not play a role.

This article proposes a new multi-commodity flow model for the LPP. The model minimizes a combination of total passenger traveling time and operating costs. It generates lines dynamically, handles frequencies implicitly by means of continuous frequency variables, and allows passengers to change their routes according to the computed line system; in particular, we do not assume a system split. These properties aim at line planning scenarios in public transport, where we see less justification for a system split and fewer restrictions in line design than one seems to have in railway line planning.

This paper is organized as follows. Section 2 gives an overview of the literature on the LPP. Section 3 introduces and discusses our model. Section 4 investigates aspects of a column generation solution approach for the LP relaxation of the model. We show that the pricing problem for the passenger variables is a shortest path problem, while the pricing problem for the lines turns out to be an $\mathcal{NP}$-hard longest path problem. However, if only lines of logarithmic length with respect to the number of nodes are considered, the pricing problem can be solved in polynomial time. In Section 5, computational results of an implementation on a practical problem for the city of Potsdam, Germany, are reported.

The goal of this article is to show that such an extended model is tractable and can be used to optimize the line plan of a medium sized town.

2 Related Work

This section provides a short overview of the literature for the line planning problem. More information can be found in the survey article of Ceder and Israeli [7], which covers the literature up to the beginning of the 1990ies; see also Odoni, Rousseau, and Wilson [18] and Bussieck, Winter, and Zimmermann [6].

The first approaches to the line planning problem had the idea to assemble lines from shorter pieces in an iterative (and often interactive) process. An early example is the so-called skeleton method described by Silman, Barzily, and Passy [21], that chooses the endpoints of a route and several intermediate nodes which are then joined by shortest paths with respect to length or traveling time; for a variation see Dubois, Bel, and Llibre [12]. In a similar way, Sonntag [22] and Pape, Reinecke, and Reinecke [19] constructed lines by adjoining small pieces of streets/tracks in order to maximize the number of direct travelers.

Another branch of the literature considers two-step approaches that precompute some set of lines in a first phase and choose a line plan from this set in a second phase. For example, Wilson [8] described an enumeration method to generate lines whose length is within a certain factor from the length of the shortest path, while Mandl [17] proposed a local search strategy to optimize over such a set. Ceder and Israeli [7, 16] introduced a quadratic set covering approach.

An important line of developments is based on the concept of the so-called *system split*. Starting point is a classification of the transportation system into levels of different speed, as common in railway systems. Assuming that travelers are likely to change to fast levels as early and leave them as late as possible, the passengers are distributed onto several paths in the system, using Kirchhoff-like rules at the transit points, before any lines are known. Note that this fixes, in particular, the passenger flow on each individual link in the network. The system split was promoted by Bouma and Oltrogge [2], who used it to develop a branch-and-bound based software system for the planning and analysis of the line system of the Dutch railway network.

Recently, advanced integer programming techniques have been applied to the line planning problem. Bussieck, Kreuzer, and Zimmermann [4] (see also Bussieck [3]) and Claessens, van Dijk, and Zwaneveld [9] both propose cut-and-branch approaches to select lines from a previously generated set of potential lines and report computations on real world railway data. Both articles deal with homogeneous transport systems, which can be assumed after a system-split is performed as a preprocessing step. Bussieck, Lindner, and Lübbecke [5] extend this work by incorporating nonlinear components into the model. Goossens, van Hoesel, and Kroon [14, 15] show that practical railway problems can be solved within reasonable quality and time by a branch-and-cut approach, even for the simultaneous optimization of several transportation systems.

3 Line Planning Model

We typeset vectors in bold face, scalars in normal face. If $\boldsymbol{v} \in \mathbb{R}^J$ is a real valued vector and I a subset of J, we denote by $\boldsymbol{v}(I)$ the sum over all components of $\boldsymbol{v}$ indexed by I, i.e., $\boldsymbol{v}(I) := \sum_{i \in I} v_i$.

For the line planning problem (LPP) we are given an undirected multigraph $G = (V, E) = (V, E_1 \dot\cup \ldots \dot\cup E_k)$, a number k of transportation modes (bus, tram, subway, etc.), terminal sets $\mathcal{T}_1, \ldots, \mathcal{T}_k \subseteq V$, operating costs $\boldsymbol{c}^1 \in \mathbb{Q}_+^{E_1}, \ldots, \boldsymbol{c}^k \in \mathbb{Q}_+^{E_k}$ on the edges, fixed costs $C_1, \ldots, C_k \in \mathbb{Q}_+$, vehicle capacities $\kappa_1, \ldots, \kappa_k \in \mathbb{Q}_+$ for each mode, and edge capacities $\boldsymbol{\lambda} \in \mathbb{Q}_+^E$. Denote by $G_i = (V, E_i)$ the subgraph of G corresponding to mode i.

A *line* of mode i is a path in G_i connecting two (different) terminals of $\mathcal{T}_i$. Note that paths are always *simple*, i.e., the repetition of nodes is not allowed. Let $c_\ell := \sum_{e\in\ell} c_e^i$ be the operating cost of line ℓ of mode i, $C_\ell := C_i$ be its fixed cost, and $\kappa_\ell := \kappa_i$ be its vehicle capacity. Let $\mathcal{L}$ be the set of all lines. Furthermore, $\mathcal{L}_e := \bigcup\{\ell \in \mathcal{L} : e \in \ell\}$ is the set of lines that use edge $e \in E$.

The problem formulation involves a (not necessarily symmetric) origin-destination matrix (OD-matrix) $(d_{st}) \in \mathbb{Q}_+^{V\times V}$ of travel demands, i.e., d_{st} is the amount of passengers that want to travel from node s to t. Let $D := \{(s,t) \in V \times V : d_{st} > 0\}$ be the set of all *OD-pairs*.

We consider a directed *passenger route graph* (V, A) derived from $G = (V, E)$ by replacing each edge $e \in E$ with two antiparallel arcs $a(e)$ and $\overline{a}(e)$. Let $e(a) \in E$ be the undirected edge corresponding to $a \in A$. For simplicity of notation, we denote this digraph also by $G = (V, A)$. We are given *traveling times* $\tau_a \in \mathbb{Q}_+$ for every arc $a \in A$. For an OD-pair $(s,t) \in D$, an (s,t)-passenger path is a directed path in (V, A) from s to t. Let $\mathcal{P}_{st}$ be the set of all (s,t)-passenger paths, $\mathcal{P} := \bigcup\{p \in \mathcal{P}_{st} : (s,t) \in D\}$ the set of all passenger paths, and $\mathcal{P}_a := \bigcup\{p \in \mathcal{P} : a \in p\}$ the set of all passenger paths that use arc a. The *traveling time* of a passenger path p is defined as $\tau_p := \sum_{a\in p} \tau_a$.

With this notation, the line planning problem can be modeled using three kinds of variables:

$y_p \in \mathbb{R}_+$: the flow of passengers traveling from s to t on path $p \in \mathcal{P}_{st}$,
$x_\ell \in \{0, 1\}$: a decision variable for using line $\ell \in \mathcal{L}$,
$f_\ell \in \mathbb{R}_+$: frequency of line $\ell \in \mathcal{L}$.

The model now reads:

$$
\text{(LPP)} \quad \min \ \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}
$$

$$
\begin{aligned}
\boldsymbol{y}(\mathcal{P}_{st}) &= d_{st} && \forall\, (s,t) \in D && \text{(i)}\\
\boldsymbol{y}(\mathcal{P}_a) - \sum_{\ell : e(a)\in\ell} \kappa_\ell f_\ell &\le 0 && \forall\, a \in A && \text{(ii)}\\
\boldsymbol{f}(\mathcal{L}_e) &\le \lambda_e && \forall\, e \in E && \text{(iii)}\\
\boldsymbol{f} &\le F\boldsymbol{x} && && \text{(iv)}\\
x_\ell &\in \{0,1\} && \forall\, \ell \in \mathcal{L} && \text{(v)}\\
f_\ell &\ge 0 && \forall\, \ell \in \mathcal{L} && \text{(vi)}\\
y_p &\ge 0 && \forall\, p \in \mathcal{P}. && \text{(vii)}
\end{aligned}
$$

The *passenger flow constraints* (i) and the nonnegativity constraints (vii) model a multi-commodity flow problem for the passenger flow, where the commodities correspond to the OD-pairs $(s,t) \in D$. This part guarantees that the demand is satisfied. The *capacity constraints* (ii) link the passenger paths with the line paths to ensure sufficient transportation capacities on each arc. The *frequency constraints* (iii) bound the total frequency of lines using each edge. Inequalities (iv) link the frequency with the decision variables for the use of lines; they guarantee that the frequency of a line is 0 whenever it is not used. Here, F is an upper bound on the frequency of a line; for technical reasons, we assume that $F \geq \lambda_e$ for all $e \in E$, see Section 4 for more information.

3.1 Discussion of the Model

Let us point out some properties of the model before we investigate its complexity.

Objectives: The objective of the model has two competing parts, namely, to minimize costs $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}$ and to minimize total passenger traveling time $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$. Here, $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x}$ is the fixed cost for setting up lines and $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}$ is the variable cost for operating these lines at frequencies $\boldsymbol{f}$. The model allows to adjust the relative importance of one part over the other by an appropriate scaling of the respective objective coefficients.

OD-Matrices: Each entry in an OD-matrix gives the number of passengers that want to travel from one point in the network to another point within a fixed time horizon. It is well known that such data have certain deficiencies. For instance, OD-matrices depend on the geometric discretization used, they are highly aggregated, they give only a snapshot type of view, it is often questionable how well the entries represent the real situation, and they should only be used when the transportation demand is fixed. However, OD-matrices currently are industry standard for estimating transportation demand. It is already quite an art and rather costly to assemble this data and there is currently no alternative in sight.

Time horizon: In the LPP, the time horizon comes into play implicitly via the OD-matrix. Usually, such data are aggregated over one day, but it is similarly appropriate to aggregate, for instance, over the rush hours. In fact, the asymmetry of demands in rush hours was one of the reasons to consider directed passenger paths.

Passenger Routes: Since the traveling times $\boldsymbol{\tau}$ are nonnegative, we can assume passenger routes to be (simple) paths.

Our model does not fix passenger paths according to a system split, but allows to freely route passengers in the line network. This is targeted at local public transport systems, where, in our opinion, people determine their traveling paths according to the line system and not only according to the network topology. To our knowledge, such routings have not been considered in the context of line planning before.

Note that the collection of passenger paths minimizes the total traveling times $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ in the sense of a system optimum. In this case, with a linear objective function and capacities, the system optimum is also a user equilibrium, namely the so-called Beckmann user equilibrium. See Correa, Schulz, and Stier Moses [10] for more information. We do, however, not address the question why passengers should choose this equilibrium out of several possible equilibria that can arise in routing with capacities.

The routing in our model allows for passengers paths of arbitrary lengths, which may force some passengers to long detours. We remark that this problem could be handled by introducing some type of resource constraints. However, this would turn the pricing problem for the passenger paths into an $\mathcal{NP}$-hard resource constrained shortest path problem; see Section 4.2. Note that such an approach models a range of path lengths with respect to the underlying network, not with respect to the computed line system, which is what one would really like to achieve.

Line Routes: The literature generally takes line routes as (simple) bidirected paths, and we do the same in this article. In fact, a restriction forcing some sort of simplicity is necessary in order to prevent repetitions around cycles; see Section 4.3. As a slight generalization of the concept of simplicity, one could investigate the case where one assumes that every line route is bounded in length and "almost" simple, i.e., no node is repeated within a given (fixed) interval.

In principle, it is easy to incorporate additional constraints on the formation of individual lines, as well as constraints on sets of lines, e.g., bounds on the number of edges in a line, or that the length of line should not deviate too much from a shortest path between its endpoints. Such constraints are important in practice.

Transfers: Transfers between lines are currently ignored in our model, because constraints (iii) only control the total capacity on edges and not the assignment of passengers to lines. The problem here are not transfers between different modes, which can be handled by linking the mode networks G_i with appropriate transfer edges, weighted by estimated transfer times. A similar trick could in principle be used for transfers between lines of the same mode, using an appropriate expansion of the graph. However, this greatly increases the complexity of the model, and it introduces degeneracy; it is unclear whether such a model remains tractable for practical data.

Frequencies: Frequencies indicate the (approximate) number of times vehicles are employed to serve the demand over the time horizon. In a real world line plan, frequencies have to produce a regular timetable and hence are not allowed to take arbitrary fractional values. Our model, however, treats frequencies as continuous values. This is a simplification. We could have forced our model to accept only a finite number of frequencies by enumerating lines with fixed frequencies, in a similar way as Claessens, van Dijk, and Zwaneveld [9] and Goossens, van Hoesel, and Kroon [14, 15]; but this would greatly increase the complexity of our model. However, as the frequencies are mainly used to adjust line capacities, we do (at present) not care so much about "nice" frequencies and view the fractional values as approximations or clues to "sensible" values.

4 LP Relaxation

The LP relaxation of (LPP) can be simplified by eliminating the $\boldsymbol{x}$-variables. In fact, since (LPP) minimizes over nonnegative costs, one can assume w.l.o.g. that the inequalities (iv) are satisfied with equality, i.e., there is an optimal LP solution such that $Fx_\ell = f_\ell \Leftrightarrow x_\ell = f_\ell/F$ for all lines ℓ. Eliminating $\boldsymbol{x}$ from the system using these equations and setting $\gamma_\ell = C_\ell/F + c_\ell$, we arrive at the following equivalent, but simpler, LP:

$$
\text{(LP)} \qquad \min\ \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{f}
$$

$$
\begin{aligned}
\boldsymbol{y}(\mathcal{P}_{st}) &= d_{st} && \forall\,(s,t)\in D && \text{(i)}\\
\boldsymbol{y}(\mathcal{P}_a) - \sum_{\ell:e(a)\in\ell} \kappa_\ell f_\ell &\le 0 && \forall\,a\in A && \text{(ii)}\\
\boldsymbol{f}(\mathcal{L}_e) &\le \lambda_e && \forall\,e\in E && \text{(iii)}\\
f_\ell &\ge 0 && \forall\,\ell\in\mathcal{L} && \text{(iv)}\\
y_p &\ge 0 && \forall\,p\in\mathcal{P}. && \text{(v)}
\end{aligned}
$$

Since $F \ge \lambda_e$, the inequalities $f_\ell \le F$ remaining after the elimination are dominated by inequalities (iii) and can be omitted. Hence, (LP) contains only a polynomial number of inequalities (apart from the nonnegativity constraints (iv) and (v)).

We aim at solving (LP) with a column generation approach and therefore investigate the corresponding pricing problems. These pricing problems are studied in terms of the dual of (LP). Denote the variables of the dual as follows: $\pi = (\pi_{st}) \in \mathbb{R}^D$ (flow constraints (i)), $\mu = (\mu_a) \in \mathbb{R}^A$ (capacity constraints (ii)), and $\eta \in \mathbb{R}^E$ (frequency constraints (iii)). The dual of (LP) is:

$$
\text{(DLP)} \qquad \max\ \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\pi} - \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\eta}
$$

$$
\begin{aligned}
\pi_{st} - \boldsymbol{\mu}(p) &\le \tau_p && \forall\,p\in\mathcal{P}_{st},\ (s,t)\in D\\
\kappa_\ell\,\boldsymbol{\mu}(\ell) - \boldsymbol{\eta}(\ell) &\le \gamma_\ell && \forall\,\ell\in\mathcal{L}\\
\boldsymbol{\mu},\ \boldsymbol{\eta} &\ge 0,
\end{aligned}
$$

where

$$
\boldsymbol{\mu}(\ell) = \sum_{e\in\ell} \big(\mu_{a(e)} + \mu_{\overline{a}(e)}\big).
$$

It will turn out that the pricing problem for the line variables f_ℓ is a longest path problem; the pricing problem for the passenger variables y_p, however, is a shortest path problem.

4.1 Complexity of the LP Relaxation

Proposition 4.1. *The computation of the optimal value of (LP) is $\mathcal{NP}$-hard in the strong sense, even for planar graphs.*

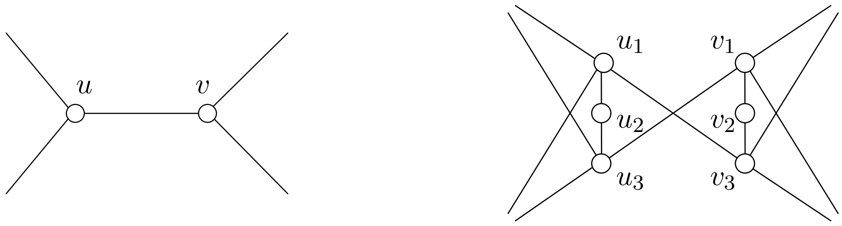

Figure 1: Example for the node splitting in the proof of Proposition 4.1

Proof. We reduce the Hamiltonian path problem, which is strongly $\mathcal{NP}$-complete even for planar graphs [13], to the LPP. Let (H, s, t) be an instance of the Hamiltonian path problem, i.e., $H = (V, E)$ is a graph and s and t are two distinct nodes of H.

For the reduction, we are going to derive an appropriate instance of (LP). The underlying network is formed by a graph $H' = (V', E')$, which arises from H by splitting each node v into three copies v_1, v_2, and v_3. For each node $v \in V$, we add edges $\{v_1, v_2\}$ and $\{v_2, v_3\}$ to E' and for each edge $\{u, v\}$ the edges $\{u_1, v_3\}$ and $\{u_3, v_1\}$, see Figure 1. Our instance of (LP) contains just a single mode with only two terminals s_1 and t_3 such that every line must start at s_1 and end at t_3. The demands are $d_{v_1 v_2} = 1$ $(v \in V)$ and 0 otherwise, and the capacity of every line is 1. For every $e \in E$, we set λ_e to some high value (e.g., to $|V|$). The cost of all edges is set to 0, except for the edges in $\delta(s_1)$, for which the costs are set to 1. The traveling times are set to 0 everywhere. It follows that the value of a solution to (LP) is the sum of the frequencies of all lines.

Assume that $p = (s, v^1, \ldots, v^k, t)$ (for $v^1, \ldots, v^k \in V$) is an (s, t)-Hamiltonian path in H. Then $p' = (s_1, s_2, s_3, v_1^1, v_2^1, v_3^1, \ldots, v_1^k, v_2^k, v_3^k, t_1, t_2, t_3)$ is an (s_1, t_3)-Hamiltonian path in H', which gives rise to an optimal solution of (LP). Namely, we can take p' as the route of a single line with frequency 1 and route the demands $d_{v_1 v_2} = 1$ for every $v \in V$ on this line directly from v_1 to v_2. As the frequency of p' is 1, the objective value of this solution is also 1. On the other hand, every solution to (LP) must have value at least one, as every line has to pass an edge of $\delta(s_1)$ and the sum of the frequencies of lines visiting an arbitrary edge of type $\{v_1, v_2\}$, for $v \in V$, is at least 1. This proves that (LP) has a solution of value 1 if (H, s, t) contains a Hamiltonian path.

For the converse, assume that there exists a solution to (LP) of value 1, for which we ignore lines with frequency 0. We know that every edge $\{v_1, v_2\}$ $(v \in V)$ is covered by at least one line of the solution. If every line contains all edges $\{v_1, v_2\}$ $(v \in V)$, each such line gives rise to a Hamiltonian path (since the line paths are simple) and we are done. Otherwise, there must be an edge $e = \{v_1, v_2\}$ $(v \in V)$ which is not covered by all of the lines. By the capacity constraints (ii), the sum of the frequencies of the lines covering e is at least 1. However, the edges in $\delta(s_1)$ are covered by the lines covering edge e plus at least one more line of nonzero frequency. Hence, the total sum of all frequencies is larger than one, which is a contradiction to the assumption that the solution has value 1.

This shows that there exists an (s,t)-Hamiltonian path in H if and only if the value of (LP) with respect to H' is 1. □

4.2 Pricing of the Passenger Variables

The reduced cost $\overline{\tau}_p$ for variable y_p for $p \in \mathcal{P}_{st}$, $(s,t) \in D$, is

$$\overline{\tau}_p = \tau_p - \pi_{st} + \boldsymbol{\mu}(p) = \tau_p - \pi_{st} + \sum_{a \in p} \mu_a = -\pi_{st} + \sum_{a \in p} (\mu_a + \tau_a).$$

The pricing problem for the $\boldsymbol{y}$-variables is to find a path p such that $\overline{\tau}_p < 0$ or to conclude that no such path exists. This can easily be done in polynomial time as follows. For all $(s,t) \in D$, we search for a shortest (s,t)-path with respect to the nonnegative weights $(\mu_a + \tau_a)$ on the arcs; we can, for instance, use Dijkstra's algorithm. If the length of this path is less than π_{st}, then y_p is a candidate variable to be added to the LP, otherwise we proved that no such path exists (for the pair (s,t)). Note that each passenger path can assumed to be simple: just remove cycles of length 0 – or trust Dijkstra's algorithm, which produces only simple paths.

4.3 Pricing of the Line Variables

The pricing problem for line variables f_ℓ is more complicated. The reduced cost $\overline{\gamma}_\ell$ for a variable f_ℓ is

$$\overline{\gamma}_\ell = \gamma_\ell - \kappa_\ell \, \boldsymbol{\mu}(\ell) + \boldsymbol{\eta}(\ell) = \gamma_\ell - \sum_{e \in \ell} \big(\kappa_\ell \, (\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e\big).$$

The corresponding pricing problem consists in finding a (simple) path ℓ of mode i such that

$$\begin{aligned} 0 > \overline{\gamma}_\ell &= \gamma_\ell - \textstyle\sum_{e \in \ell} \big(\kappa_\ell \, (\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e\big) \\ &= C_\ell / F + c_\ell - \textstyle\sum_{e \in \ell} \big(\kappa_\ell \, (\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e\big) \\ &= C_i / F + \textstyle\sum_{e \in \ell} c_e^i - \sum_{e \in \ell} \big(\kappa_i \, (\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e\big) \\ &= C_i / F + \textstyle\sum_{e \in \ell} \big(c_e^i - \kappa_i \, (\mu_{a(e)} + \mu_{\overline{a}(e)}) + \eta_e\big) \\ \Leftrightarrow \; \textstyle\sum_{e \in \ell} (\kappa_i \, &(\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e - c_e^i) > C_i / F. \end{aligned}$$

This problem turns out to be a maximum weighted path problem, since the weights $(\kappa_i \, (\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e - c_e^i)$ are not restricted in sign. Hence, the pricing problem for the line variables is $\mathcal{NP}$-hard (even for planar graphs) [13].

This scenario, however, may be a little too pessimistic. In fact, better algorithmic properties can be achieved when assuming bounds on the lengths of the lines, i.e., the number of edges used in a line. Let us start with two arguments why this case has some relevance.

The first argument is based on an idea of a transportation network as a planar graph, probably of high connectivity. Suppose this network occupies a square, in which n nodes are evenly distributed. A typical line starts in the outer regions of the network, passes through the center, and ends in another outer region; we would expect such a line to be of length $\mathcal{O}(\sqrt{n})$.

Real networks, however, are not only (more or less) planar, but often resemble trees. In a *balanced* and preprocessed tree, where each node degree is at least 3, the length of a path between any two nodes is only $\mathcal{O}(\log n)$.

We now provide a result which shows that the maximum weighted path problem can be solved in polynomial time in the case when the length of a path is logarithmic in the number of nodes. Our result is a direct generalization of work by Alon, Yuster, and Zwick [1]. Our method works both for directed and undirected graphs.

Consider for this purpose the graph $G = (V, E)$ (with only one mode) with arbitrary edge weights $w_e \in \mathbb{Q}$ for all $e \in E$ and choose a source node s. We let $n = |V|$ and $m = |E|$. The problem is to compute a maximum weight path with respect to $\boldsymbol{w}$ in G from s to all other nodes of length $\mathcal{O}(\log n)$.

The goal of the work of Alon et al. is to find induced paths of fixed length $k - 1$ in a graph. The basic idea is to randomly color the nodes of the graph with k colors and only allow paths that use distinct colors for each node; such paths are called *colorful* with respect to the coloring and are necessarily simple. Choosing a coloring $c : V \to \{1, \ldots, k\}$ uniformly at random, every path using at most $k - 1$ edges has a chance of at least $k!/k^k > e^{-k}$ to be colorful with respect to c. If we repeat this process $\alpha \cdot e^k$ times with $\alpha > 0$, the probability that a given path p with at most $k - 1$ edges is never colorful is less than

$$\left(1 - e^{-k}\right)^{\alpha \cdot e^k} < e^{-\alpha}.$$

Hence, the probability that p is colorful at least once is at least $1 - e^{-\alpha}$. The search for such colorful paths can be performed using dynamic programming, which leads to an algorithm running in $m \cdot 2^{\mathcal{O}(k)}$ expected time. This algorithm is then derandomized.

This argument yields the following result, which can easily be generalized to directed graphs.

Proposition 4.2. *Let $G = (V, E)$ be a graph, let k be a fixed number, and $c : V \to \{1, \ldots, k\}$ be a coloring of the nodes of G. Let s be a node in G and (w_e) be edge weights. Then a colorful maximum weight path with respect to $\boldsymbol{w}$ using at most $k - 1$ edges from s to every other node can be found in time $\mathcal{O}(m \cdot k \cdot 2^k)$, if such paths exist.*

Proof. We find the maximum weight of such paths by dynamic programming. Let $v \in V$, $i \in \{1, \ldots, k\}$, and $C \subseteq \{1, \ldots, k\}$ with $|C| \leq i$. Define

$w(v, C, i)$ to be the weight of the maximum weight colorful path with respect to $\boldsymbol{w}$ from s to v using at most $i-1$ edges and using the colors in C. Hence, for each iteration i we store the set of colors of all maximum weight colorful paths from s to v using at most $i - 1$ edges. Note that we do not store the set of paths, only their colors. Hence, at each node we store at most 2^i entries. The entries of the table are initialized with minus infinity and we set $w(s, \{c(s)\}, 1) = 0$.

At iteration $i \geq 1$, let (u, C, i) be an entry in the dynamic programming table. If for some edge $e = \{u, v\} \in E$ we have $c(v) \notin C$, let $C' = C \cup \{c(v)\}$ and set

$$w(v, C', i + 1) = \max \big\{w(u, C, i) + w_a,\ w(v, C', i + 1),\ w(v, C', i)\big\}.$$

The term $w(v, C', i + 1)$ accounts for the cases where we already found a path to v (using at most i edges) with higher weight, whereas $w(v, C', i)$ makes sure that paths using at most $i - 1$ edges to v are accounted for. After iteration $i = k$, we take the maximum of all entries corresponding to each node v, which is the wanted result. The number of updating steps is bounded by

$$\sum_{i=0}^{k} i \cdot 2^i \cdot m = m \cdot \big(2 + 2^{k+1}(k - 1)\big) = \mathcal{O}\big(m \cdot k \cdot 2^k\big).$$

The sum on the left side of this equation arises as follows. In iteration i, m edges are considered; each edge $\{u, v\}$ starts at node u, to which at most 2^i labels $w(u, C, i)$ are associated, one for each possible set C; for each such set, checking whether $c(v) \in C$ takes time $\mathcal{O}(i)$. The summation formula itself can be proved by induction, see also [20, Exc. 5.7.1, p. 95]. The algorithm can be easily modified to actually find the maximum weight paths. □

We can now use Proposition 4.2 to produce an algorithm which finds a maximum weight path in $\alpha\, e^k\, \mathcal{O}\big(mk2^k\big) = \alpha\, \mathcal{O}\big(m \cdot 2^{\mathcal{O}(k)}\big)$ time with high probability. Then a derandomization can be performed by a clever enumeration of colorings such that each path with at most $k-1$ edges is colorful with respect to at least one such coloring. Alon et al. combine several techniques to show that $2^{\mathcal{O}(k)} \cdot \log n$ colorings suffice. Applying this result we obtain:

Theorem 4.1. *Let $G = (V, E)$ be a graph and let k be a fixed number. Let s be a node in G and (w_e) be edge weights. Then a maximum weight path with respect to $\boldsymbol{w}$ using at most $k - 1$ edges from s to every other node can be found in time $\mathcal{O}\big(m \cdot 2^{\mathcal{O}(k)} \cdot \log n\big)$, if such paths exist.*

If $k \in \mathcal{O}(\log n)$, this yields a polynomial time algorithm. Hence, by the discussion above, it follows that the LP relaxation (LP) of the line planning problem can be solved in polynomial time in this case. On the other hand we have following result.

Proposition 4.3. *It is $\mathcal{NP}$-hard to compute a maximum weight path of length at most k, if $k \in \mathcal{O}(n^{1/N})$ for any fixed $N \in \mathbb{N} \setminus \{0\}$.*

Proof. Consider an instance (H, s, t) for the Hamiltonian path problem, where H is a graph with n nodes. We add $(n^N - n)$ isolated nodes to H in order to obtain the graph H' with n^N nodes, which is polynomial in n. Let the weights on the edges be 1. If we would be able to find a maximum weight path with at most $k = (n^N)^{1/N} = n$ edges starting from s, we could solve the Hamiltonian path problem for H. □

5 Computational Experience

In this section we report on computational experience with line planning problems for the city of Potsdam, Germany. The experiments originate from a joint project with two local public transportation companies in Potsdam (ViP Verkehrsgesellschaft GmbH and Havelbus Verkehrsgesellschaft mbH), the city of Potsdam, and the software company IVU Traffic Technologies AG.

Potsdam is a medium sized town near Berlin; it has about 150,000 inhabitants. Its public transportation system uses city buses and trams (operated by ViP) and regional busses (operated by Havelbus). Additionally, there are regional trains connecting Potsdam to its surroundings (operated by Deutsche Bahn AG) and a city railroad (operated by S-Bahn Berlin) which provides connections to Berlin. As regional trains and the city railroad are not operated by ViP and Havelbus, the associated lines routes and frequencies were assumed to be fixed.

5.1 Data

Our data consists of a multi modal traffic network of Potsdam and an associated OD-matrix, which had been used by IVU in a consulting project for the Potsdam line network (Nahverkehrsplan). The data represents the line system of Potsdam of 1998 and has 27 bus lines and 4 tram lines. Including line variants, the total number of lines was 80. The original network had 951 nodes, including 111 OD-nodes, and 1321 edges. This data was preprocessed as follows.

We removed isolated nodes. Then we iteratively removed "leaves" in the graph, i.e., nodes which have only one neighbor, and iteratively contracted nodes with two neighbors. We remark that although such preprocessing steps are conceptually easy, the data handling can be quite intricate in practice; for instance, our data included information on possible turnings of a line at road/rail crossings, which must be updated in the course of the preprocessing.

The OD-matrix was also modified. Nodes with no traffic were removed. The original time horizon was one day, but we scaled the matrix to 40% in an (admittedly rough) attempt to simulate afternoon traffic (3 p.m. to 6 p.m.). Note that the resulting matrix still is quite symmetric (the maximum difference between each of the two directions was 25) whereas a real afternoon OD-matrix would not be symmetric.

The preprocessed graph had 410 nodes, 106 of which were OD-nodes, and 891 edges. The scaled OD-matrix had 4685 nonzeros and the total scaled traveling demand was 42796.

As an overall objective, we used

$$\min \; \lambda\,(\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}) + (1-\lambda)\,\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}, \tag{1}$$

where $\lambda \in [0,1]$ is a parameter weighting the two parts. The traveling time is measured in seconds. Since no data was available on line costs, we decided on $C_\ell = 10000$ (fixed costs) for each line ℓ and $c_e^i = 100$ (operating costs) for each edge e and mode i. Hence, we do not distinguish between costs of different modes (an unrealistic assumption in practice).

5.2 Algorithms

We have implemented a column generation algorithm based on the model (LPP). The algorithm solves the LP relaxation in the first phase and constructs a feasible line plan using a greedy type heuristic.

For column generation, our implementation iteratively prices out passenger and line paths until no improving variables are found. The master LPs are solved with the barrier algorithm and, towards the end, with the primal simplex algorithm of CPLEX 9.0.

Passenger paths are priced out by using Dijkstra's algorithm, as explained in Section 4.2 . This pricing part is very fast, but sometimes many iterations were necessary to find the correct paths.

The pricing of the (simple) line paths was performed by a restricted enumeration, which works as follows. For each mode i and each pair (a,b) of terminals from $\mathcal{T}_i$, where $a \neq b$, we enumerated lines of length ≤ 50; this restriction arises from the fact that the maximal length of the original lines in our data was 47 (in the preprocessed network). We furthermore restricted the line paths as follows. We computed the minimal number $k(a,b)$ of edges needed to connect a and b in G_i. We only allowed lines with $k \leq \max\{\alpha \cdot k(a,b), 50\}$ edges, where we used $\alpha = 1.2$. The idea is to produce only lines that do not deviate too much from a shortest path. This pricing step is also quite fast; the reason seems to be that the mode graphs G_i for this Potsdam network are almost "tree-like", i.e., they include few cycles and typically have low degree (3 or 4).

Our heuristic for generating an integer solution of (LPP) is motivated by the observation that the solution of the LP relaxation (LP) contains many

Table 1: Reference solution and optimized solution of (LP) and integer solution of (LPP) computed by the greedy method for the case $\lambda = 0.9977$.

Reference solution:		
total passenger time:	103,922,094.00	[scaled: 239,020.82]
total line cost:	506,105.26	[scaled: 504,941.22]
LP objective value:	743,962.04	
active line/pass. var.:	68/4896	
Optimized solution:		
total passenger time:	107,628,512.75	[scaled: 247,545.58]
total line cost:	253,913.16	[scaled: 253,329.16]
LP objective value:	500,874.74	
active line/pass. var.:	67/4889	
Integer solution of greedy method:		
total passenger time:	115,125,491.50	[scaled: 264,788.63]
total line cost:	317,757.46	[scaled: 317,026.61]
LP objective value:	581,815.24	integer: 950,964.24
active line/pass. var.:	37/4772	

lines with very low frequencies; such lines are not reasonable for practice. We try to remove these lines by a simple greedy method. It removes lines in the order of increasing frequencies as long as the remaining set of lines still is feasible, i.e., all demands can be transported. The $\boldsymbol{x}$-variables of the resulting lines are set to 1 and all other $\boldsymbol{x}$-variables to 0.

5.3 Experiments

We report results of several computational experiments with this data and implementation. All experiments were performed on a 3 GHz Pentium 4 machine running Linux.

Let us point out explicitly that we do not claim that these results are already practically significant; we only want to show that there is a potential to apply our methods to practical data.

In our first experiment, we solved the LP relaxation (LP) for $\lambda = 0.9977$, which roughly balances the two parts of the objective function. The resulting problem had 5773 rows. After 152 iterations (i.e., solutions of the master LP) and 371 seconds we obtained an optimal solution shown in the middle part of Table 1. We performed 7 pricing rounds for lines. The pricing needed a total time of 60 seconds of which most was used for the pricing of line paths. Hence, most of the time is spent in solving LPs.

The reference solution, shown in the upper part of Table 1, was computed by fixing the paths of the original lines of Potsdam and then solving the resulting LP relaxation without generating new lines, but allowing the

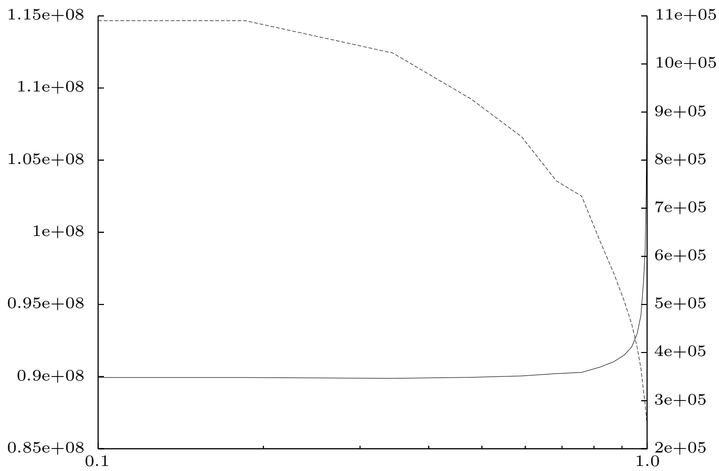

Figure 2: Total traveling time (solid, left axis) and total line cost (dashed, right axis) in dependence on λ (x-axis in logscale).

frequencies of the lines to change. Comparing these two solutions in Table 1 shows that allowing the generation of new line paths reduces the line cost to roughly 50% and the total objective to roughly 67% of the original values.

We also investigated the passenger routing of our LP solution. To connect the 4685 OD-pairs only 4889 paths are needed, i.e., most OD-pairs are connected by a unique path. The total traveling time in the system is 10,758,179 seconds; weighting with the number of passengers for each connection gives 107,628,512.75, i.e., the total passenger time listed in Table 1. In comparison, when we route all passengers between every OD-pair on the fastest path in the final line system, the total traveling time is 8,308,002 seconds and weighted with the number of passengers: 92,938,324. This is a relative difference of 23% for the total traveling time and of 14% when weighted with the number of passengers. This seems to be an acceptable deviation.

Our second experiment investigates the dependence of the solution on the parameter λ. We computed the solutions to the LP relaxation for 21 different values of λ_i, taking $\lambda_i = 1 - \big(1 - i/20\big)^4$, for $i = 0, \ldots, 20$. This collects increasingly more samples near $\lambda = 1$, a region where the total passenger traveling time and the total line cost are about equal.

The results are plotted in Figure 2. This figure shows the total traveling time and the total line cost depending on λ. The extreme cases are as expected: For $\lambda = 0$, the line costs do not contribute to the objective and are therefore high, while the total traveling time is low. For $\lambda = 1$, only the total line cost contributes to the objective and is therefore minimized as much as possible at the cost of increasing the total traveling time. With increasing λ, the total line cost monotonically decreases, while the total traveling time increases. Note that each computed pair of total traveling time and line cost constitutes a Pareto optimal point, i.e., is not dominated by any other attainable combination.

In our third experiment, we computed an integer solution for (LPP) associated with the parameter $\lambda = 0.9977$. The application of our greedy algorithm results in a solution with 37 lines, down from 67 from our first solution. (Here only lines among city buses, trams, and regional busses could be removed.) The final objective without fixed costs is 581,815.24, with total passenger time 115,125,491.50 seconds, and the total line cost is 317,757.46; the fixed costs are $\lambda \cdot 370,000 = 369,149$, see Table 1.

We can compare this solution to the integer solution obtained by rounding all nonzero $\boldsymbol{x}$-variables in the solution of the LP relaxation to 1. The objective of the rounded solution including fixed costs is 1,158,642.15, which leads to a gap of 56.83% to the value of the LP-relaxation. For the solution computed by the Greedy heuristic the integer objective is 950,964.24 and the gap is 38.82%. This gap seems reasonable, keeping in mind that LP-relaxations of fixed cost problems typically produce bad lower bounds and some of the lines were not allowed to be removed. Still, more research is needed to provide better lower bounds by integer programming methods and to produce better primal solutions.

Acknowledgement. We thank Volker Kaibel for pointing out Proposition 4.3.

References

[1] N. Alon, R. Yuster, and U. Zwick, *Color-coding*, J. Assoc. Comput. Mach. **42**, no. 4 (1995), pp. 844–856.

[2] A. Bouma and C. Oltrogge, *Linienplanung und Simulation für öffentliche Verkehrswege in Praxis und Theorie*, Eisenbahntechnische Rundschau **43**, no. 6 (1994), pp. 369–378.

[3] M. R. Bussieck, *Optimal lines in public rail transport*, PhD thesis, TU Braunschweig, 1997.

[4] M. R. Bussieck, P. Kreuzer, and U. T. Zimmermann, *Optimal lines for railway systems*, Eur. J. Oper. Res. **96**, no. 1 (1997), pp. 54–63.

[5] M. R. Bussieck, T. Lindner, and M. E. Lübbecke, *A fast algorithm for near optimal line plans*, Math. Methods Oper. Res. **59**, no. 2 (2004).

[6] M. R. Bussieck, T. Winter, and U. T. Zimmermann, *Discrete optimization in public rail transport*, Math. Program. **79B**, no. 1–3 (1997), pp. 415–444.

[7] A. Ceder and Y. Israeli, *Scheduling considerations in designing transit routes at the network level*, in Proc. of the Fifth International Workshop on Computer-Aided Scheduling of Public Transport (CASPT), Montréal, Canada, 1990, M. Desrochers and J.-M. Rousseau, eds., Lecture Notes in Economics and Mathematical Systems 386, Springer-Verlag, Berlin, Heidelberg, 1992, pp. 113–136.

[8] A. Ceder and N. H. M. Wilson, *Bus network design*, Transportation Res. **20B**, no. 4 (1986), pp. 331–344.

[9] M. T. Claessens, N. M. van Dijk, and P. J. Zwaneveld, *Cost optimal allocation of rail passanger lines*, Eur. J. Oper. Res. **110**, no. 3 (1998), pp. 474–489.

[10] J. R. Correa, A. S. Schulz, and N. E. Stier Moses, *Selfish routing in capacitated networks*, Math. Oper. Res. **29** (2004), pp. 961–976.

[11] J. R. Daduna, I. Branco, and J. M. P. Paixão, eds., *Proc. of the Sixth International Workshop on Computer-Aided Scheduling of Public Transport (CASPT), Lisbon, Portugal, 1993*, Lecture Notes in Economics and Mathematical Systems 430, Springer-Verlag, Berlin, Heidelberg, 1995.

[12] D. Dubois, G. Bel, and M. Llibre, *A set of methods in transportation network synthesis and analysis*, J. Oper. Res. Soc. **30**, no. 9 (1979), pp. 797–808.

[13] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[14] J.-W. H. M. Goossens, S. van Hoesel, and L. G. Kroon, *On solving multi-type line planning problems*, METEOR Research Memorandum RM/02/009, University of Maastricht, 2002.

[15] J.-W. H. M. Goossens, S. van Hoesel, and L. G. Kroon, *A branch-and-cut approach for solving railway line-planning problems*, Transportation Sci. **38**, no. 3 (2004), pp. 379–393.

[16] Y. Israeli and A. Ceder, *Transit route design using scheduling and multi-objective programming techniques*, in Daduna et al. [11], pp. 56–75.

[17] C. E. Mandl, *Evaluation and optimization of urban public transportation networks*, Eur. J. Oper. Res. **5** (1980), pp. 396–404.

[18] A. R. Odoni, J.-M. Rousseau, and N. H. M. Wilson, *Models in urban and air transportation*, in Handbooks in OR & MS 6, S. M. Pollock et al., ed., North Holland, 1994, ch. 5, pp. 107–150.

[19] U. Pape, Y.-S. Reinecke, and E. Reinecke, *Line network planning*, in Daduna et al. [11], pp. 1–7.

[20] M. Petkovsek, H. S. Wilf, and D. Zeilberger, $A = B$, A. K. Peters, Wellesley, MA, 1996.

[21] L. A. Silman, Z. Barzily, and U. Passy, *Planning the route system for urban buses*, Comput. Oper. Res. **1** (1974), pp. 201–211.

[22] H. Sonntag, *Ein heuristisches Verfahren zum Entwurf nachfrageorientierter Linienführung im öffentlichen Personennahverkehr*, Z. Oper. Res. **23** (1979), pp. B15–B31.