Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

RALF BORNDÖRFER MARTIN GRÖTSCHEL MARC E. PFETSCH

A Column-Generation Approach to Line Planning in Public Transport

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

ZIB-Report 05-18 (March 2005, Original Title: "A Path-Based Model for Line Planning in Public Transport", revised November 2006)

A Column-Generation Approach to 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 involves 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 and the passengers want to minimize traveling times. We propose a new multicommodity flow model for line planning. Its main features, in comparison to existing models, are that the passenger paths can be freely routed and lines are generated dynamically. We discuss properties of this model, investigate its complexity, and present a column-generation algorithm for its solution. Computational 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*. Each step can be supported by operations research methods, see for instance the survey articles of Odoni, Rousseau, and Wilson [20] and of Bussieck, Winter, and Zimmermann [7].

This article is about the *line-planning problem* (LPP) in public transport. The problem is to design line routes and their frequencies in a street or track network such that a transportation volume, given by a so-called *origin-destination matrix* (OD-matrix), can be routed. 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* (that travel without transfers) is often considered as the objective. In such an approach, transfer waiting times do not play a role.

This article proposes a new, extended multicommodity flow model for the LPP. The model minimizes a combination of total passenger traveling time and operating costs. It generates line routes dynamically, handles frequencies 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, in which we see less justification for a system split and fewer restrictions in line design than one seems to have in railway line planning. The goal of this article is to show that such a model is tractable and can be used to optimize the line plan of a medium-sized town.

The paper is organized as follows. Section 2 surveys the literature on the LPP. Section 3 introduces and discusses our model. Section 4 presents a column-generation solution approach. 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 on a practical problem for the city of Potsdam, Germany, are reported. We end with conclusions in Section 6.

2 Related Work

This section provides a short overview of the literature for the line-planning problem. Additional information can be found in the article of Ceder and Israeli [8], which covers the literature up to the beginning of the 1990s; see also Odoni, Rousseau, and Wilson [20] and Bussieck, Winter, and Zimmermann [7].

The first approaches to the line-planning problem had the idea to assemble lines from short pieces in an iterative (and often interactive) process. An early example is the so-called skeleton method described by Silman, Barzily, and Passy [25], 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 [13]. In a similar way, Sonntag [26] and Pape, Reinecke, and Reinecke [21] constructed lines by adjoining small pieces of streets/tracks to maximize the number of direct travelers.

Successive approaches precompute some set of lines in a first phase and choose a line plan from this set in a second phase; all articles discussed in the remainder of this section use this idea. For example, Ceder and Wilson [9] described an enumeration method to generate lines whose length is within a certain factor from the length of the shortest path, while Mandl [19] proposed a local search strategy to optimize over such a set. Ceder and Israeli [8, 18] introduced a quadratic set covering approach.

An important line of developments is based on the concept of the so-called *system split*. Its starting point is a classification of the links of a transportation system into levels of different speed, as is 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. This fixes the passenger flow on each individual link in the network. The system split was promoted by Bouma and Oltrogge [3], 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 [5] (see also Bussieck [4]) and Claessens, van Dijk, and Zwaneveld [10] 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 [6] extend this work by incorporating nonlinear components into the model. Goossens, van Hoesel, and Kroon [16, 17] show that practical railway problems can be solved within reasonable time and quality by a branch-and-cut approach, even for the simultaneous optimization of several transportation systems. Schöbel and Scholl [23, 24] study a Dantzig-Wolfe decomposition approach to route passengers through an expanded line-network to minimize the number of transfers or the transfer time.

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 a number M of transportation *modes* (bus, tram, subway, etc.), an undirected multigraph $G = (V, E) = (V, E_1 \dot\cup \ldots \dot\cup E_M)$ representing a multimodal transportation network, *terminal sets* $\mathcal{T}_1, \ldots, \mathcal{T}_M \subseteq V$ of nodes for each mode where lines can start and end, line *operating costs* $\boldsymbol{c}^1 \in \mathbb{Q}_+^{E_1}, \ldots, \boldsymbol{c}^M \in \mathbb{Q}_+^{E_M}$ on the edges, *fixed costs* $C_1, \ldots, C_M \in \mathbb{Q}_+$ for the set-up of a line for each mode, *ve-*

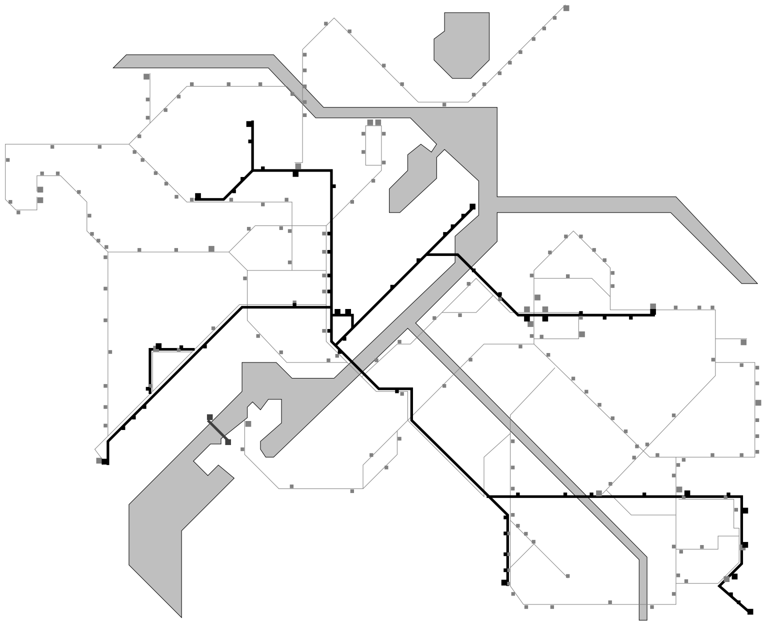

Figure 1: Multimodal transportation network in Potsdam. Black: tram, light gray: bus, dark gray: ferry, large nodes: terminals, small nodes: stations, grey: rivers and lakes.

hicle capacities $\kappa_1, \ldots, \kappa_M \in \mathbb{Q}_+$ for each mode, and *edge capacities* $\mathbf{\Lambda} \in \mathbb{Q}_+^E$. Denote by $G_i = (V, E_i)$ the subgraph of G corresponding to mode i. See Figure 1 for an example network and Table 1 for a list of notation that we use throughout the paper.

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; it is possible to consider additional constraints on the formation of lines such as a maximum length, etc. 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 feasible 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 further 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 number of passengers who want to travel from node s to node t. Let $D := \{(s,t) \in V \times V \,:\, d_{st} > 0\}$ be the set of all *OD-pairs*.

Finally, we derive a directed *passenger route graph* (V, A) from $G = (V, E)$ by replacing each edge $e \in E$ with two antiparallel arcs $a(e)$ and $\overline{a}(e)$; conversely, 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

Table 1: Notation and terminology.

G	multimodal transport network	G_i	subnetwork for mode i
$\mathcal{T}_i$	terminals for mode i	$\boldsymbol{c}^i$	line operating costs for mode i
c_ℓ	operating costs for line ℓ	C_i	line fixed costs for mode i
κ_i	vehicle capacity for mode i	κ_ℓ	vehicle capacity for line ℓ
$\mathcal{L}$	set of all lines	$\mathcal{L}_e$	lines using edge e
D	set of OD-pairs	d_{st}	travel demand between s and t
τ_a	traveling time on arc a	τ_p	traveling time on path p
$\mathcal{P}$	set of all passenger paths	$\mathcal{P}_{st}$	paths between s and t
y_p	passenger flow on path p	x_ℓ	whether line ℓ is used
f_ℓ	frequency of line ℓ	Λ_e	frequency bounds for edge e

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 LPP can be modeled using three kinds of variables:

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

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

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

The *passenger flow constraints* (i) and the nonnegativity constraints (vii) model a multicommodity 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 routed. The *capacity constraints* (ii) link the passenger paths with the line paths to ensure sufficient transportation capacity on each arc. The *frequency constraints* (iii) bound the total frequency of lines using an edge. Inequalities (iv) link the frequencies with the decision variables for the use of lines; they guarantee that the frequency of a line is zero 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.

Let us discuss some properties of the model before we investigate its algorithmic tractability.

Objectives: The objective of the model has two competing parts, namely, to minimize total passenger traveling time $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ and to minimize costs $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}$. 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. Including fixed costs allows to consider objectives such as minimizing the number of lines; note that LPP is a linear program (LP) if all fixed costs are zero.

OD-Matrices: Each entry in an OD-matrix gives the number of passengers who 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 can be assumed to be fixed. However, OD-matrices are at present the industry standard for estimating transportation demand. It is already quite an art and rather costly to assemble this data, and currently, no alternative is in sight.

Time horizon: The LPP implicitly contains a time horizon via the OD-matrix. Usually, OD-data are aggregated over one day, but it is similarly appropriate to consider, for instance, peak traffic in rush hours. In fact, the asymmetry of demands in rush hours was one of the reasons why we consider directed passenger paths.

Passenger Routes: Because 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 according to the computed lines. 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. Except for the work of Schöbel and Scholl [23, 24], which is independent of ours, such routings have not been considered in the context of line planning before.

Our model computes a set of passenger paths that minimize the total traveling times $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ in the sense of a system optimum. However, in our case, with a linear objective function and linear capacities, it can be shown that the resulting system optimum is also a user equilibrium, namely, the so-called Beckmann user equilibrium, see Correa, Schulz, and Stier Moses [11]. We do not address the question of 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 travel times, which may force some passengers to long detours. We remark that this problem could be handled by introducing appropriate bounds on the travel times of paths. This would, however, turn the pricing problem for the passenger paths into an $\mathcal{NP}$-hard resource-constrained shortest path problem; see Section 4.1. Note also that such an approach would measure travel times with respect to shortest paths in the underlying network (independent of any line system). Ideally, however, one would like to compare to the shortest paths using only arcs covered by the computed line system.

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 to prevent repetitions around cycles. As a slight generalization of the concept of simplicity, one could investigate the case in which one assumes that every line route is bounded in length or "almost" simple, i.e., no node is repeated within a given interval.

It is easy to incorporate additional constraints on the formation of individual lines and constraints on sets of lines, e.g., that the length of a line should not deviate too much from a shortest path between its endpoints or bounds on the number of lines using an edge. Such constraints are important in practice. In this article we consider bounds on the number of edges in a line. Let us give two arguments why this case is practically relevant.

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 $O(\sqrt{n})$.

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

Transfers: Transfers between lines are currently ignored in our model, because constraints (ii) only control the total capacity on edges and not the assignment of passengers to lines. The problem 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. In principle, a similar trick could 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 need to be employed to serve the demand over the time horizon. In a real-world line plan, frequencies often 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 have introduced fixed costs to reduce the number of lines and decrease the likelihood of low frequencies. In addition, 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 [10] and Goossens, van Hoesel, and Kroon [16, 17]; but the resulting model would be much harder to solve. However, as the frequencies mainly are 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 Column Generation

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 inequalities (iv) above 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. Substituting for $\boldsymbol{x}$, we observe that the inequalities $f_\ell \le F$ remaining after the elimination are dominated by inequalities (iii) and, hence, can be omitted (recall that we assumed $F \ge \Lambda_e$). Setting $\gamma_\ell = C_\ell/F + c_\ell$, we arrive at the following equivalent, but simpler, linear program:

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

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

Note that (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 (see Barnhart et al. [2] for an introduction) 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: $\boldsymbol{\pi} = (\pi_{st}) \in \mathbb{R}^D$ (flow constraints (i)), $\boldsymbol{\mu} = (\mu_a) \in \mathbb{R}^A$ (capacity constraints (ii)), and $\boldsymbol{\eta} \in \mathbb{R}^E$ (frequency constraints (iii)). The dual of (LP) is:

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

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 Pricing of the Passenger Variables

The reduced cost $\overline{\tau}_p$ for variable y_p with $p \in \mathcal{P}_{st}$, $(s,t) \in D$, 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 easily can be done in polynomial time as follows. For all $(s,t) \in D$, we search for a shortest (s,t)-path p 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 p 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 we can assume that each passenger path is simple: just remove cycles of length 0 – or trust Dijkstra's algorithm, which produces only simple paths.

4.2 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

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

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

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

This problem turns out to be a maximum weighted path problem, because the weights $(\kappa_i \left(\mu_{a(e)} + \mu_{\overline{a}(e)}\right) - \eta_e - c_e^i)$ are not restricted in sign. Hence, the pricing problem for the line variables is $\mathcal{NP}$-hard [15]. This shows that solving the LP relaxation (LP) is $\mathcal{NP}$-hard as well. In fact, we can prove the stronger result that the line-planning problem itself is $\mathcal{NP}$-hard, even with fixed costs zero, independent of the model (Proposition 4.1 implies that (LP) is $\mathcal{NP}$-hard, because (LPP) is equivalent to (LP) for fixed costs 0).

Proposition 4.1. *The line-planning problem LPP is $\mathcal{NP}$-hard, even with fixed costs 0.*

Proof. We reduce the Hamiltonian path problem, which is strongly $\mathcal{NP}$-complete [15], to the LPP with fixed costs 0. 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.

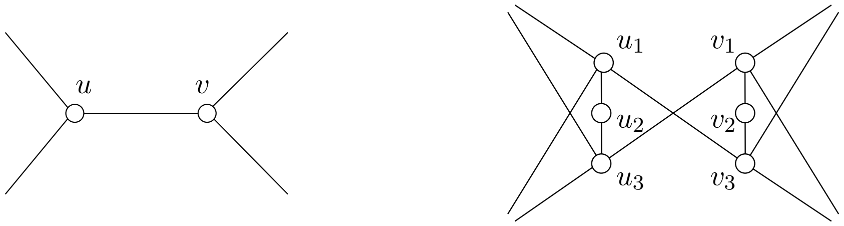

Figure 2: Example for the node splitting gadget in the proof of Proposition 4.1

For the reduction, we are going to derive an appropriate instance of LPP. 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 2. Our instance of LPP 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 incident to 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 LPP 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 LPP. 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 LPP must have value at least one, because every line has to pass an edge incident to 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 LPP has a solution of value 1, if (H, s, t) contains a Hamiltonian path.

For the converse, assume that there exists a solution to LPP 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 (because the line paths are simple) and we are done. Otherwise, there must be an edge $e = \{v_1, v_2\}$ $(v \in V)$ that is not covered by all of the lines. Because the lines have to provide enough capacity, the sum of the frequencies of the lines covering e is at least 1. However, the edges incident to 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 an optimal solution of LPP with respect to H' has value 1. $\square$

4.3 Pricing of Length Restricted Lines

Let us now consider the pricing problem for line-planning problems with bounds on the lengths of the lines, i.e., the number of edges of a line. Consider for this purpose the graph $G = (V, E)$ (for simplicity of notation with only one mode) with arbitrary edge weights $w_e \in \mathbb{Q}$ for all $e \in E$, and a source node s and a sink node t. We let $n = |V|$ and $m = |E|$. In this setting, the line-pricing problem is to find a maximum weight path from s to t with respect to $\boldsymbol{w}$. We first show that this problem is $\mathcal{NP}$-hard for the case in which the length of a line is bounded by $O(\sqrt{n})$.

Proposition 4.2. *It is $\mathcal{NP}$-hard to compute a maximum weight path from s to t of length at most k, if $k \in O\big(n^{1/N}\big)$ for any fixed $N \in \mathbb{N} \setminus \{0\}$.*

Proof. Let (H, s, t) be an instance of 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 a graph H' with n^N nodes; note that n^N is polynomial in n for fixed N. Let the weights on the edges be 1. If we could find a maximum weight path from s to t with at most $k = (n^N)^{1/N} = n$ edges in polynomial time, we could solve the Hamiltonian path problem for H in polynomial time. $\square$

We now provide a result that shows that the maximum weighted path problem can be solved in polynomial time in the case when the lengths of the paths are at most $O(\log n)$. Our method is a direct generalization of work by Alon, Yuster, and Zwick [1] on the unweighted case; it works both for directed and undirected graphs.

Alon et al. consider the problem to find simple paths of fixed length $k-1$ in a graph. Their 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

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

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.

These arguments yield the following result for the weighted undirected case, which is easily seen to be valid for directed graphs as well.

Proposition 4.3. *Let $G = (V, E)$ be a graph with m edges, k be a fixed number, and $c : V \to \{1, \dots, k\}$ 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 $O\big(m \cdot k \cdot 2^k\big)$, if such paths exist.*

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

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

The term $w(v, C', i+1)$ accounts for the cases in which 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 $O(i)$. The summation formula itself can be proved by induction, see also [22, Exc. 5.7.1, p. 95]. The algorithm can be easily modified to actually find the maximum weight paths. $\square$

We can use Proposition 4.3 to produce an algorithm that finds a maximum weight path in $\alpha\, e^k\, O\big(mk2^k\big) = \alpha\, O\big(m \cdot 2^{O(k)}\big)$ 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 $2^{O(k)} \cdot \log n$ colorings suffice. Applying this result we obtain the following.

Theorem 4.1. *Let $G = (V, E)$ be a graph with n nodes and m edges and 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 $O\big(m \cdot 2^{O(k)} \cdot \log n\big)$, if such paths exist.*

If $k \in O(\log n)$, this yields a polynomial time algorithm. Hence, by the discussion above, we get the following result.

Corollary 4.1. *The LP relaxation of (LPP) can be solved in polynomial time, if the lengths of the lines are most k, with $k \in O(\log n)$.*

4.4 Algorithm

We used the results of the previous sections to implement a column-generation algorithm for the solution of the model (LPP) with length-restricted lines. As an overall objective function, we used the weighted sum

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

where $\lambda \in [0, 1]$ is a parameter weighing the two parts.

The algorithm solves the LP relaxation in a first phase and constructs a feasible line plan using a greedy type heuristic in a second phase.

To solve the LP relaxation, our algorithm iteratively prices out passenger and line path variables until no improving variables are found. We solve the master LP with the barrier algorithm and, toward the end of the process, with the primal simplex algorithm of CPLEX 9.1. We check for new passenger path variables for all OD-pairs using Dijkstra's algorithm, see Section 4.1, until no more improving passenger paths are found. If we do not find an improving passenger path, we price out line variables for all line modes and all feasible terminal pairs. We have implemented two different methods for the pricing of (simple) line paths, namely, we either use an enumeration or the randomized coloring algorithm of Section 4.3 (we do not derandomize the algorithm). If an improving passenger or line path has been found, another iteration is started; otherwise, the LP is solved.

In the second phase, our algorithm tries to construct a good integer solution from a line pool consisting of the lines having nonzero frequencies in the optimal LP solution. The heuristic is motivated by the observation that the solution of the LP relaxation of a line-planning problem often contains lines with very low frequencies. We try to remove these lines by a simple greedy method based on a strong branching selection criterion. In the beginning, the $\boldsymbol{x}$-variables of all lines in the pool are set to 1. In each iteration, we tentatively remove a line (set its x-variable to 0), compute the objective $\lambda\, \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f} + (1-\lambda)\, \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ of the LP obtained by fixing the line variables as described, pricing passenger variables as needed, and add the fixed costs $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x}$ of all lines that are fixed to 1. After probing candidate lines with the smallest $\boldsymbol{f}$-values in this way, we permanently delete the line whose removal resulted in the smallest objective. We repeat this elimination as long as the remaining set of lines is still feasible, i.e., all demands can be routed, and the objective function decreases.

5 Computational Results

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 the two local public transport companies, 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 buses (operated by Havelbus). Additionally, regional trains connect Potsdam to its surroundings (operated by Deutsche Bahn AG) and a city railroad (operated by S-Bahn Berlin) provides connections to Berlin. Because regional trains and the city railroad are not operated by ViP and Havelbus, the associated lines routes are assumed to be fixed.

5.1 Data

Our data consists of a multimodal traffic network of Potsdam and an associated OD-matrix, which had been used by IVU in a consulting project for planning the Potsdam network (Nahverkehrsplan). The data represents the 1998 line system of Potsdam. It has 27 bus lines and 4 tram lines. Including line variants, the total number of lines was 80. The network has 951 nodes, including 111 OD-nodes, and 1,321 edges. The maximum length of a line is 47 edges.

The network was preprocessed as follows. We removed isolated nodes. Then, we iteratively removed "leaves" in the graph—i.e., nodes with only one neighbor—and iteratively contracted nodes with two neighbors. The preprocessed graph has 410 nodes, 106 of which were OD-nodes, and 891 edges. 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 zero traffic were removed. The original time horizon was one day, but we wanted to construct a line plan for the peak hour. We therefore 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 is still quite symmetric (the maximum difference between each of the two directions was 25) whereas a real afternoon OD-matrix would not be symmetric. The scaled OD-matrix had 4685 nonzeros and the total scaled travel demand was 42796.

All traveling times are measured in seconds and we always restricted the maximum length of a line to 55 edges. Because 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 Experiments

Table 2 reports the results of several computational experiments with the data and implementation we have described. All experiments were performed on a 3.4 GHz Pentium 4 machine running Linux. In the table, the *total traveling time* is $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ and *total line cost* is $\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{f}$, the *scaled* values are $(1-\lambda)\,\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ and $\lambda\,\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{f}$, respectively; all four values refer to the LP relaxation (LP). The *LP objective value* is $\lambda\,\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{f} + (1-\lambda)\,\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$, the *integer objective value* refers to $\lambda\,(\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}) + (1-\lambda)\,\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$. The last line in each block of results gives the number of active (i.e., nonzero) line and passenger variables, and the number of passenger transfers (first number) that were needed as well as the number of transfering passengers (second number). Note that we can compute transfers from passenger routes as an afterthought, although our optimization model is currently insensitive to them.

Let us point out explicitly that we do not claim that our results are already practically significant; we only want to show that there is potential to apply our methods to practical data. For example, our costs are not realistic. Therefore, the frequencies we compute cannot be compared to ones used in practice. To allow some adaptation to our cost model, we let the frequencies of all lines be variable, in particular, the frequencies of the city railroad and regional train lines.

In our first experiment, we solved the LP relaxation (LP) of the Potsdam problem, pricing lines either by enumeration or by the randomized coloring method of Section 4.3, see top of Table 2. We set $\lambda = 0.9978$, which roughly balances the two parts of the objective function. The resulting LP had 5761 rows. Using enumeration, we obtained an optimal solution after 451 seconds and 283 iterations (i.e., solutions of the master LP), of which 15 were used to price lines. The pricing problems needed a total time of 183 seconds of

Table 2: Experimental results of line planning for $\lambda = 0.9978$.

Optimized LP solution – enumeration:		
total traveling time:	108,360,036.33	[scaled: 238,392.08]
total line cost:	233,776.86	[scaled: 233,262.55]
LP objective value:	471,654.63	
active line/pass. var.:	60/4,879	transfers: 8,777/64,607
Optimized LP solution – randomized coloring – 5 trials:		
total traveling time:	108,396,741.75	[scaled: 238,472.83]
total line cost:	239,099.73	[scaled: 238,573.71]
LP objective value:	477,046.54	
active line/pass. var.:	61/4,880	transfers: 9,143/66,546
Optimized LP solution – randomized coloring – 15 trials:		
total traveling time:	108,491,234.25	[scaled: 238,680.72]
total line cost:	237,422.50	[scaled: 236,900.17]
LP objective value:	475,580.88	
active line/pass. var.:	62/4,885	transfers: 9,387/68,049
Optimized integer solution – greedy heuristic:		
total traveling time:	112,581,291.50	[scaled: 247,678.84]
total line cost:	287,060.90	[scaled: 286,429.37]
integer objective value:	818,491.68	
active line/pass. var.:	30/4,767	transfers: 8,638/60,539
Reference LP solution:		
total traveling time:	105,269,846.00	[scaled: 231,593.66]
total line cost:	501,376.24	[scaled: 500,273.21]
LP objective value:	731,866.87	
active line/pass. var.:	61/4,857	transfers: 8,618/63,310
Reference integer solution – greedy heuristic:		
total traveling time:	106,952,869.00	[scaled: 235,296.31]
total line cost:	562,964.54	[scaled: 561,726.02]
integer objective value:	1,213,221.49	
active line/pass. var.:	44/4,814	transfers: 9,509/70,525

which most was used for the pricing of line paths. Hence, more than half the time is spent for solving the master LPs.

We repeated this experiment using the randomized coloring algorithm with 5 and 15 trials for line pricing. With 5 trials, we needed 397 master LPs and 394 seconds in total; line pricing used only 99 seconds. One can see, however, that the objective is about 1% higher than for the enumeration variant. Using 15 trials resulted in 269 master LPs and 473 seconds in total. Line pricing now uses 265 seconds, and the difference in the objective function relative to the enumeration variant is reduced to 0.8%. Hence, one can achieve a good approximation of the optimal value using randomized line pricing, although approaching the optimum solution comes at the cost of larger computation times.

We also investigated in more detail the passenger routing of our LP solution for the enumeration variant. To connect the 4,685 OD-pairs only 4,879 paths are needed, i.e., most OD-pairs are connected by a unique path. The total traveling time is 108,360,036.33 seconds, see Table 2. For comparison, when we ignore capacities and route all passengers between every OD-pair on the fastest path in the final line system, the total traveling time is 95,391,460 seconds. This relative difference of 12% seems to be an acceptable deviation.

In our second experiment, we computed two integer solutions for (LPP) associated with the parameter $\lambda = 0.9978$, as above. The first solution is obtained by rounding all nonzero $\boldsymbol{x}$-variables in the solution of the LP relaxation, computed with the enumeration variant, to 1. The (integer) objective of this rounded solution is 1,058,079.69, which leads to a gap of 55% compared to the LP relaxation value of 471,654.63. The second solution is obtained by the greedy algorithm described in Section 4.4, starting from the same LP solution (only lines for city buses, trams, and regional buses were removed). It has 30 lines (17 bus lines and 2 tram lines), down from 60 in the first solution, see Table 2; it took 1,368 seconds to compute. The final (scaled) operating costs are 286,429.37, while the final fixed costs are $\lambda \cdot 300,000 = 299,340$. The integer objective of 818,491.68 has a gap of 42% with respect to the LP relaxation value of 471,654.63. Note that the results heavily depend on the cost structure: decreasing the fixed costs automatically reduces the gap. In our context, with high fixed costs, emphasis is on reducing the number of lines (recall that the costs were artificial). The result obtained seems to be quite good, given that the original line system contained 27 bus lines and 4 tram lines; it seems unlikely that one can further reduce the number. Furthermore, the lower bound of the LP relaxations typically is very weak for such fixed-cost problems. Still, more research is needed to provide better lower bounds and primal solutions.

We compare the LP and integer solutions to "reference solutions" shown in the lower part of Table 2. The reference LP solution is obtained by fixing the paths of the original lines of Potsdam and then solving the resulting LP relaxation without generating new lines, but allowing the frequencies of the lines to change. The reference integer solution is obtained by applying the greedy heuristic to the reference LP solution. The results show that allowing the generation of new line paths reduces line costs in both cases to roughly 50% and the total objective to roughly 2/3 of the original values, while the total traveling time increases by a small percent. Hence, in these experiments, the greedy algorithm has not changed the relative improvement obtained from optimizing lines.

Our third experiment investigates the influence of the parameter λ on the solution. 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 $i = 0, \dots, 20$. This collects increasingly more samples near $\lambda = 1$, a region where the total traveling time and total line cost are about equal.

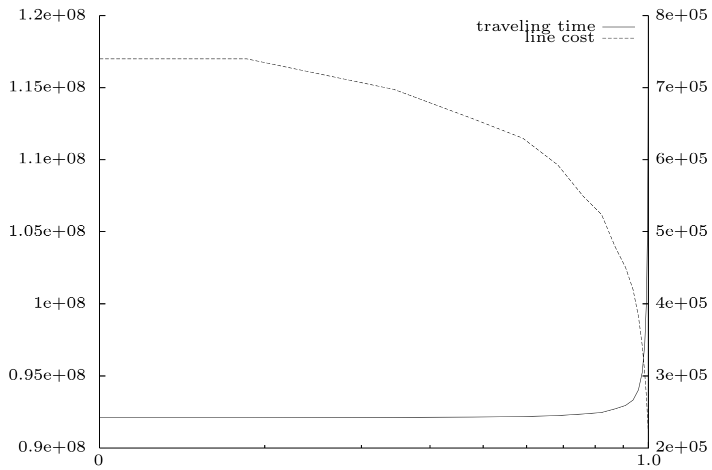

Figure 3: Total traveling time (solid, left axis) and total line cost (dashed, right axis) in dependence on λ (x-axis in logscale).

The results are plotted in Figure 3. 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. Conversely, any Pareto optimal solution of the LP relaxation can be obtained as the solution for some $\lambda \in [0, 1]$, see, e.g., Ehrgott [14].

6 Conclusions

We proposed a new model for line planning in public transport that allows to generate lines dynamically and to freely route passengers according to the computed lines. The model allows to deal with manifold requirements from practice. We showed that line-planning problems for a medium-sized town can be solved within reasonable quality with integer programming techniques. Our computational results indicate significant optimization potential. Our results on the polynomial time solvability of the LP relaxation for the case of logarithmic line lengths raises our hope that the model is suited to deal with larger problems as well.

Acknowledgment. The authors thank Volker Kaibel for pointing out Proposition 4.2.

References

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

[2] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, *Branch-and-price: Column generation for solving huge integer programs*, Oper. Res. **46**, no. 3 (1998), pp. 316–329.

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

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

[5] 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.

[6] 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).

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

[8] 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.

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

[10] 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.

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

[12] 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.

[13] 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.

[14] M. Ehrgott, *Multicriteria optimization*, Springer-Verlag, Berlin, 2nd ed., 2005.

[15] 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.

[16] 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.

[17] 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.

[18] Y. Israeli and A. Ceder, *Transit route design using scheduling and multi-objective programming techniques*, in Daduna et al. [12], pp. 56–75.

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

[20] 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.

[21] U. Pape, Y.-S. Reinecke, and E. Reinecke, *Line network planning*, in Daduna et al. [12], pp. 1–7.

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

[23] A. Schöbel and S. Scholl, *Line planning with minimal travelling time*, Tech. Report 1-2005, University of Göttingen, Germany, 2005.

[24] S. Scholl, *Customer-Oriented Line Planning*, PhD thesis, University of Göttingen, 2005.

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

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