Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

RALF BORNDÖRFER MARIKA NEUMANN

Models for Line Planning with Transfers

Supported by the DFG Research Center MATHEON “Mathematics for key technologies”, project B15.

ZIB-Report 10-11 (June 2010)

Models for Line Planning with Transfers*

Ralf Borndörfer** Marika Neumann**

June 12, 2010

Abstract

We propose a novel integer programming approach to transfer minimization for line planning problems in public transit. The idea is to incorporate penalties for transfers that are induced by "connection capacities" into the construction of the passenger paths. We show that such penalties can be dealt with by a combination of shortest and constrained shortest path algorithms such that the pricing problem for passenger paths can be solved efficiently. Connection capacity penalties (under)estimate the true transfer times. This error is, however, not a problem in practice. We show in a computational comparison with two standard models on a real-world scenario that our approach can be used to minimize passenger travel and transfer times for large-scale line planning problems with accurate results.

1 Introduction

Line planning is a classical optimization problem in the design of a public transportation system. Given an infrastructure network and an origin-destination matrix of travel demands, the line planning problem is to find a set of lines or paths in the network with corresponding operation frequencies such that all travel demands can be satisfied. There are two main objectives, namely, minimization of operation costs (the operator's point of view) and minimization of travel and transfer times (the passengers' point of view).

Since the late nineteen-nineties, the line planning literature has developed a series of integer programming models that try to capture these objectives better and better, see Odoni, Rousseau, and Wilson [16] and Bussieck, Winter, and Zimmermann [8] for an overview. Operation costs are discussed in the articles of Claessens, van Dijk, and Zwaneveld [9], Bussieck, Lindner, and Lübbecke [7], and Goossens, van Hoesel, and Kroon [12, 13]; we focus in this article on travel and transfer times. Bussieck, Kreuzer,

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

**Zuse-Institute Berlin (ZIB), Takustr. 7, 14195 Berlin, Germany, Email {borndoerfer, marika.neumann}@zib.de

and Zimmermann [6] (see also the thesis of Bussieck [5]) first proposed an integer programming model that maximizes the number of *direct travelers* (i.e., the number of travelers with zero transfers) on the basis of a "system split" of the demand, i.e., an a priori distribution of the passenger flow on the arcs of the transportation network. Schöbel and Scholl [18, 19] model travel and transfer times explicitly in terms of a change-and-go graph that is constructed on the basis of all potential lines. This model allows a complete and accurate formulation of travel and transfer time objectives; its only drawback is its size. Nachtigall and Jerosch [15] address this problem and achieve a graph reduction with a column generation approach in terms of partial passenger paths between two transfers; however, the number of rows in the associated integer programming formulation still grows with the number of lines. Borndörfer, Grötschel, and Pfetsch [3, 4] propose a computationally efficient integrated line planning and passenger routing model with a polynomial number of constraints. The disadvantage of this model is that it ignores transfers between lines of the same mode completely (transfers between, e.g., bus and tram lines are considered).

We propose in this paper a new approach that tries to combine the advantages of the models of Borndörfer, Grötschel, and Pfetsch [3, 4] and Schöbel and Scholl [18, 19]. Augmenting the passenger routing component of the first model by a term that accounts for transfers that are induced by direct connection capacities allows to minimize an (under)estimate of transfer times. Since the computational efficiency of the original model is retained, this makes progress towards a treatment of transfers that is both efficient and exact. We substantiate this claim by a computational study for a large-scale real-world line planning scenario for the city of Potsdam in Germany.

2 Line Planning Models

We state in this section three line planning models. The first is a *basic model* similar to that of Borndörfer, Grötschel, and Pfetsch [3, 4]; it does not consider transfers. The second model extends the first by accounting for (a lower bound on) the number of transfers in a passenger path. This model, henceforth referred to as *direct connection capacity model*, relates the demand for direct passenger paths to an associated transportation capacity; if the capacity for direct passenger paths is exceeded, the remaining passengers must travel on paths with one or more transfers. The direct connection capacity model overestimates the capacity for direct connections, such that some passengers paths account for less transfers than needed, i.e., the model accounts for some, but not all transfers. The third model is the *change-and-go model* of Schöbel and Scholl [18, 19]; it serves as a reference to compute the correct travel times and transfer penalties.

We use the following notation. Consider a public transportation network as a graph $N = (V, E)$, whose nodes and edges correspond to stations and connections between these stations, respectively. Denote by $\mathcal{L}$ the line pool, i.e., a set of paths in N that represent all valid lines, by $\mathcal{L}_{st}$ the set of lines that provide a direct connection between nodes s and t, by $\mathcal{F} \subseteq \mathbb{N}$ the set of possible frequencies at which these lines can be operated, by $F \in \mathbb{N}$ an upper bound on the maximum number of lines that can be operated on an edge of N, and by $\kappa_\ell \in \mathbb{Q}_+$ the vehicle capacity of line ℓ. Let further $(d_{st}) \in \mathbb{Q}_+^{V\times V}$ be an origin-destination (OD) matrix that gives the travel demand between pairs of nodes, and denote by $D = \{(s,t) \in V \times V : d_{st} > 0\}$ the set of all OD-pairs with positive demand. Derive a directed passenger routing graph $\bar{N} = (V, A)$ from N by replacing each edge $e \in E$ with two antiparallel arcs $a(e)$ and $\bar{a}(e)$. Denote by $\mathcal{P}_{(s,t)}$ the set of all possible directed (s,t)-paths in $\bar{N}$ for $(s,t) \in D$, and by $\mathcal{P} = \bigcup_{(s,t)\in D} \mathcal{P}_{(s,t)}$ the set of all such paths; these represent travel routes of passengers. Associated with each arc $a \in A$ and path $p \in \mathcal{P}$ are travel times $\tau_a \in \mathbb{Q}_+$ and $\tau_p = \sum_{a\in p} \tau_a$, respectively, and with each transfer a (uniform) penalty $\sigma \in \mathbb{Q}_+$. Let k_p be the minimum number of transfers that passengers must do on path p if all lines in $\mathcal{L}$ would be built. Let $\mathcal{K}_p = \{k_p, \ldots, |V| - 1\}$ be a set of possible numbers of transfers on path $p \in \mathcal{P}$. A path $p \in \mathcal{P}$ with $k \in \mathcal{K}_p$ transfers has travel and transfer time $\tau_{p,k} = \tau_p + k\sigma$. Let $e(a)$ be the undirected edge corresponding to $a \in A$, and let us interpret a(n undirected) line in N in such a way that passengers can travel on this line in both directions in $\bar{N}$.

Denote finally for an integer programming model M and some relaxation R by $v_R(M)$ the optimal objective value of relaxation R of M.

2.1 Basic Model

Consider the following *basic model* for line planning:

$$
\begin{array}{llll}
\text{(B)} \quad \min & \lambda \sum_{\ell\in\mathcal{L}} \sum_{f\in\mathcal{F}} c_{\ell,f}\, x_{\ell,f} + (1-\lambda) \sum_{p\in\mathcal{P}} \tau_p\, y_p & & \\
\text{(i)} & \sum_{p\in\mathcal{P}_{st}} y_p & = d_{st} & \forall\, (s,t) \in D \\
\text{(ii)} & \sum_{p: a\in p} y_p & \le \sum_{\ell: e(a)\in\ell} \sum_{f\in\mathcal{F}} \kappa_{\ell,f}\, x_{\ell,f} & \forall\, a \in A \\
\text{(iii)} & \sum_{f\in\mathcal{F}} x_{\ell,f} & \le 1 & \forall\, \ell \in \mathcal{L} \\
\text{(iv)} & \sum_{\ell: e(a)\in\ell} \sum_{f\in\mathcal{F}} f \cdot x_{\ell,f} & \le F & \forall\, e \in E \\
\text{(v)} & x_{\ell,f} & \in \{0,1\} & \forall\, \ell \in \mathcal{L},\ f \in \mathcal{F} \\
\text{(vi)} & x_{\ell,f} & \ge 0 & \forall\, \ell \in \mathcal{L},\ f \in \mathcal{F} \\
\text{(vii)} & y_p & \ge 0 & \forall\, p \in \mathcal{P}.
\end{array}
$$

Model (B) differs from the one in Borndörfer, Grötschel, and Pfetsch [3] by the use of binary variables $x_{\ell,f}$ for the operation of line $\ell \in \mathcal{L}$ at frequency $f \in \mathcal{F}$. The continuous variables y_p account for the number of passengers that travel on path $p \in \mathcal{P}$. We denote by $c_{\ell,f}$ the cost of line ℓ operated at frequency f, and by $\kappa_{\ell,f} = \kappa_\ell \cdot f$ the capacity of line ℓ operated with frequency f. The objective function of program (B) minimizes a sum of line operating costs and passenger travel times, weighted by a parameter $0 \le \lambda \le 1$. Equations (i) stipulate a passenger flow of d_{st} for each OD-pair $(s,t) \in D$. Inequalities (ii) enforce sufficient transportation capacity on each arc. Inequalities (iii) ensure that a line is operated with at most one frequency, while inequalities (iv) bound the number of lines that can be operated on an individual edge.

Program (B) contains an exponential number of inequalities (iii). These so-called *SOS constraints* (see Beale and Tomlin [2]) are usually redundant in real world line planning problems. This is due to cost structures that satisfy

$$c_{\cdot,f_i} + c_{\cdot,f_j} > c_{\cdot,f_k} \qquad \text{for all } f_i, f_j, f_k \in \mathcal{F}$$

because of the presence of a substantial fixed cost component. Such costs favor the operation of a single frequency for each line. In fact, replacing two low frequencies by a higher one only creates trouble if overcapacities get in conflict with the frequency constraints (iv). In practice, this is rarely the case. However, it is possible to construct examples where the SOS constraints are violated; such cases can be dealt with using *SOS branching*, see [17]. We therefore propose (B) (i), (ii), (iv), (vi), and (vii) as an empirically strong *SOS-LP relaxation* of program (B), that not only relaxes the integrality constraints (v), but also the SOS constraints (iii). Constraints (iii) and (v) can be resolved in the branch-and-bound process, the SOS-LP relaxation itself by column generation.

Proposition 2.1. *The pricing problem for passenger path variables in the SOS-LP relaxation of program (B) is a shortest path problem. It can be solved in polynomial time. The pricing problem for line paths in the SOS-LP relaxation of program (B) is an NP-hard longest path problem. It can be solved in polynomial time if the lines have length $O(\log |V|)$.*

Proof. The proof is similar to that in Borndörfer, Grötschel, and Pfetsch [3]. □

2.2 Direct Connection Capacity Model

The main disadvantage of the basic model is that it ignores transfers completely and therefore greatly underestimates travel disutilities. Our idea to mitigate this problem is to account for the number $k \in \mathcal{K}$ of transfers on each passenger path $p \in \mathcal{P}$, and for the transportation capacity of lines that permit the connections required by such passenger paths. Then, if there is not enough transportation capacity for paths with a small number of transfers, paths with more transfers have to be chosen. Unfortunately, such transportation capacities seem to be hard to determine. However, approximations in terms of underestimations on the number of transfers on a passenger path and, likewise, necessary conditions on transportation capacities can be derived. We describe now an implementation of this idea that focuses on direct connections.

Consider the following *direct connection capacity model*:

$$
\begin{array}{llll}
\text{(DCC)} & \multicolumn{3}{l}{\min\ \lambda \displaystyle\sum_{\ell\in\mathcal{L}}\sum_{f\in\mathcal{F}} c_{\ell,f}\, x_{\ell,f} + (1-\lambda)\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}_p} \tau_{p,k}\, y_{p,k}} \\
\text{(i)} & \displaystyle\sum_{p\in\mathcal{P}_{st}}\sum_{k\in\mathcal{K}_p} y_{p,k} & = d_{st} & \forall\,(s,t)\in D \\
\text{(ii)} & \displaystyle\sum_{p\in\mathcal{P}:a\in p}\sum_{k\in\mathcal{K}_p} y_{p,k} & \le \displaystyle\sum_{\ell\in\mathcal{L}:e(a)\in\ell}\sum_{f\in\mathcal{F}} \kappa_{\ell,f}\, x_{\ell,f} & \forall\, a\in A \\
\text{(iii)} & \displaystyle\sum_{p\in\mathcal{P}_{st}:a\in p, 0\in\mathcal{K}_p} y_{p,0} & \le \displaystyle\sum_{\ell\in\mathcal{L}_{st}:e(a)\in\ell}\sum_{f\in\mathcal{F}} \kappa_{\ell,f}\, x_{\ell,f} & \forall\, a\in A,\ \forall\,(s,t)\in D \\
\text{(iv)} & \displaystyle\sum_{f\in\mathcal{F}} x_{\ell,f} & \le 1 & \forall\,\ell\in\mathcal{L} \\
\text{(v)} & \displaystyle\sum_{\ell\in\mathcal{L}:e(a)\in\ell}\sum_{f\in\mathcal{F}} f\cdot x_{\ell,f} & \le F & \forall\, e\in E \\
\text{(vi)} & x_{\ell,f} & \in\{0,1\} & \forall\,\ell\in\mathcal{L},\ f\in\mathcal{F} \\
\text{(vii)} & x_{\ell,f} & \ge 0 & \forall\,\ell\in\mathcal{L},\ f\in\mathcal{F} \\
\text{(viii)} & y_{p,k} & \ge 0 & \forall\, p\in\mathcal{P},\ k\in\mathcal{K}_p.
\end{array}
$$

Model (DCC) differs from model (B) by the use of continuous variables $y_{p,k}$ that account for the number of passengers that travel on path $p \in \mathcal{P}$ doing *at least* k transfers; let us call k the *transfer estimate* of path p. By definition, the objective coefficients of the passenger path variables satisfy $\tau_{p,j} \le \tau_{p,k}$ for $j < k$, and the model will use the smallest transfer estimate, ideally $k = 0$, unless forced otherwise. Such an enforcement is given by constraints (viii) and (iii). The *minimum transfer* constraints (viii) stipulate the theoretical lower bound on the transfer estimate. Note that paths that cannot be direct connections account for a transfer penalty of at least $k_p\sigma$ that is ignored in model (B). The *direct connection capacity* constraints (iii) bound the volume of direct (s,t)-travelers on an arc a by the transportation capacity of lines that connect s and t. If this capacity is exceeded, some (s,t)-paths must take at least one transfer that would also have been ignored by model (B). Of course, some capacity of the lines $\mathcal{L}_{st}$ on arc a might be used up by other travelers, i.e., the right hand side of constraint (iii) overestimates the capacity for direct (s,t)-connections. It can therefore happen that the model calculates zero transfer estimates on some passenger paths that actually require at least one transfer. The remaining constraints are similar to model (B): Equations (i) enforce the passenger flow, inequalities (ii) sufficient transportation capacity on each arc, inequalities (iv) are the SOS frequency constraints, and inequalities (v) bound the total frequency on each arc.

Proposition 2.2. *Model DCC dominates model B as an integer program and with* *respect to the SOS-LP relaxation, i.e.,* $v_{IP}(DCC) \geq v_{IP}(B)$ *and* $v_{SOS-LP}(DCC) \geq v_{SOS-LP}(B)$.

Proof. $y_p = \sum_{k \in \mathcal{K}_p} y_{p,k}$ is feasible for (B) for any solution of (DCC). □

Consider now the solution of the SOS-LP relaxation of program (DCC) by a column generation approach. To this purpose, associate dual variables π, $\mu \geq 0$, $\nu \geq 0$, and $\eta \geq 0$ with constraints (i), (ii), (iii), and (v) of program (DCC). The pricing problem for line variables is

$$\min_{\ell \in \mathcal{L},\, f \in \mathcal{F}} \lambda c_{\ell,f} - \sum_{e \in \ell} \left[\eta_e + \sum_{a \in \{a(e), \bar{a}(e)\}} \left(\mu_a + \sum_{(s,t) \in D} \nu_{a,(s,t)} \right) \right]. \tag{1}$$

Except for the numerical values on the edges, this problem is identical to the line pricing problem for the basic model (B). The pricing problems for the passenger variables $y_{p,0}$ and $y_{p,k}$, $k \geq 1$, are as follows:

$$\min_{(s,t) \in D} -\pi_{(s,t)} + \min_{p \in \mathcal{P}_{(s,t)}} \min_{k \in \mathcal{K}_p \cap \{0\}} \sum_{a \in p} \left[(1-\lambda)\tau_a + \mu_a + \nu_{a,(s,t)} \right] \tag{2a}$$

$$\min_{(s,t) \in D} -\pi_{(s,t)} + \min_{p \in \mathcal{P}_{(s,t)}} \min_{k \in \mathcal{K}_p \setminus \{0\}} \sum_{a \in p} \left[(1-\lambda)\tau_a + \mu_a \right] + (1-\lambda)k\sigma. \tag{2b}$$

These problems can be solved using shortest path techniques. Consider for a line $\ell \in \mathcal{L}$ the directed path $\ell_{(u,v)}$ that connects u to v in $\bar{N} = (V, A)$ via arcs $a(e)$, $e \in \ell$, where $\ell_{(u,v)} = \emptyset$ if $u \notin \ell$ or $v \notin \ell$. For any $u, v \in V$ let

$$\delta^{\omega}_{uv} = \min_{\ell \in \mathcal{L}: u,v \in \ell} \sum_{a \in \ell_{(u,v)}} \omega_a \tag{3}$$

be the length, possibly ∞, of a shortest direct connection between u and v in $\bar{N} = (V, A)$ on lines in $\mathcal{L}$ with respect to arc lengths $\omega \geq 0$. Suppose that the *direct connection length* matrix (δ^{ω}_{uv}) can be computed for arbitrary arc weights $\omega \geq 0$ in time $O(\Phi)$.

Consider first the pricing problem (2a) for direct (s,t)-passenger paths. Setting $\omega_a = (1-\lambda)\tau_a + \mu_a + \nu_{a,(s,t)}$ for all $a \in A$, problem (2a) is equivalent to

$$\min_{(s,t) \in D} -\pi_{(s,t)} + \delta^{\omega}_{st}.$$

Consider now the pricing problem (2b) for (s,t)-passenger paths with at least one transfer. Setting $\omega_a = (1-\lambda)\tau_a + \mu_a$ for all $a \in A$,

$$\theta^{\omega}_{uv,st} = \delta^{\omega}_{uv} + (1-\lambda)\sigma + \begin{cases} \infty, & uv = st, \\ 0, & \text{else}, \end{cases}$$

Algorithm 1: Computing shortest direct connections in a transportation network via length constrained lines between a set of terminals.

Input : Transportation network $G = (V, E)$ with edge weights $\omega \geq 0$, set of terminals $T \subseteq V$, length bound $K \in \mathbb{N}$

Output: Shortest direct connection matrix (δ^ω_{uv}) w.r.t. edge weights ω

for all $u, v \in V$, $u \neq v$, $k = 0, \ldots, K$ **do**
  $d^k_{uv} = \infty$
end
for all $v \in V$, $k = 0, \ldots, K$ **do**
  $d^k_{vv} = 0$, $\delta^\omega_{vv} = 0$
end
for all $u \in V$ **do**
  for $k = 1, \ldots, K$ **do**
    for all $(v, w) \in A$ **do**
      $d^k_{uw} = \min\{d^{k-1}_{uv} + \omega_{vw}, d^{k-1}_{uw}\}$
    end
  end
end
for $u, v \in V$, $u \neq v$ **do**
  $\delta^\omega_{uv} = \min\{d^j_{uv} : \exists s, t \in T, \exists i, j, k \in \mathbb{N} : d^i_{su} + d^j_{uv} + d^k_{vt} < \infty, i + j + l \leq K\}$
end

for $u, v \in V$, and Δ^ω_{st}, $(s, t) \in D$, to the length of a shortest (s, t)-path in a complete digraph with node set V and arc lengths $\theta^\omega_{uv,st}$ (note that $\theta^\omega_{st,st} = \infty$ such that a shortest (s, t)-path will use at least two direct connection arcs), problem (2b) is equivalent to

$$\min_{(s,t)\in D} -\pi_{(s,t)} - (1 - \lambda)\sigma + \Delta^\omega_{st}.$$

Problem (2a) can be solved in time $O(|D|\Phi)$, problem (2b) in time $O(\Phi + |D|\phi(|V|))$, where $\phi(|V|)$ is the time needed to solve a single source shortest path problem on a complete digraph with $|V|$ nodes.

It remains to investigate the computation of the shortest direct connection matrix (δ^ω_{uv}). This computation depends on the encoding of the line pool. If the line pool is given explicitly, the matrix entries can be computed by enumeration via Equation (3). If the line pool is generated dynamically, shortest direct connection times can be computed in a similar way as the lines themselves. We give here an example for *length constrained lines* as discussed in Borndörfer, Grötschel, and Pfetsch [3]; other line construction rules can be dealt with analogously.

Suppose $T \subseteq V$ is a set of terminals between which lines can be constructed. Let us stipulate that a line contains at most some number K of edges. Then (δ^ω_{uv}) can be computed by Algorithm 1 in time $\Phi = O(|T|^2|V|^2K^2)$, and the pricing problem (2) can be solved in $O(|D||T|^2|V|^2K^2) \leq O(|V|^6K^2)$. The complexity of line pricing is $O(|T|^2|E|K2^K) \leq O(|V|^4K2^K)$, see Borndörfer, Grötschel, and Pfetsch [3], for the basic and for the direct connection capacity model. The complexity of the passenger pricing problem is polynomial for the basic and pseudo-polynomial for the direct connection capacity model. As line pricing is the hard part, the direct connection capacity model retains the computational complexity of the basic model.

Proposition 2.3. *Consider a line planning problem with length constrained lines. The pricing problem for passenger path variables in the SOS-LP relaxation of program (DCC) is a constraint shortest path problem. It can be solved in pseudo-polynomial time. The pricing problem for line paths in the SOS-LP relaxation of program (B) is an NP-hard longest path problem. It can be solved in polynomial time if the lines have length* $O(\log |V|)$.

2.3 Change-and-Go Model

Schöbel and Scholl [18] proposed an approach that allows for a correct modeling of transfers. The idea is to set up a so-called change-and-go network that contains the nodes and edges of all lines, i.e., the change-and-go network contains a copy of each node and edge for every line that contains this nodes and edge, respectively. Further transfer edges are added to model possible transfers. This approach aims at small, explicitly computable line pools.

A formal statement of the change-and-go model is as follows. The basis is a *change-and-go graph* $G = (\mathcal{V}, \mathcal{E})$. Its nodes $\mathcal{V} = \mathcal{V}_O \cup \mathcal{V}_L$ represent either pairs of stations and lines or origin/destination nodes, i.e.,

$$\begin{aligned} \mathcal{V}_L &= \{(s, \ell) \mid s \in V\ \ell \in \mathcal{L}\}, \\ \mathcal{V}_O &= \{s \in V : \exists\, t \in V,\ d_{st} + d_{ts} > 0\}. \end{aligned}$$

Its edges $\mathcal{E} = \mathcal{E}_O \cup \mathcal{E}_L \cup \mathcal{E}_T$ connect OD-nodes to the network, different nodes of the same line (traveling edges), and different nodes at the same station (transfer edges):

$$\begin{aligned} \mathcal{E}_O &= \{\big(s, (s, \ell)\big) \mid s \in V,\ \ell \in \mathcal{L} : s \in \ell\} \\ \mathcal{E}_L &= \{\big((u, \ell), (v, \ell)\big) \mid \ell \in \mathcal{L} : (u, v) \in E\} \\ \mathcal{E}_T &= \{\big((v, \ell), (v, \tilde{\ell})\big) \mid \ell \in \mathcal{L},\ \tilde{\ell} \in \mathcal{L}\}. \end{aligned}$$

Similar as for the basic model, we consider a directed passenger routing graph $D = (\mathcal{V}, \mathcal{A})$ derived from G by replacing each edge with two antiparallel arcs. We associate with the line arcs $\mathcal{A}_L$ and the OD arcs $\mathcal{A}_O$ the travel times of the corresponding arcs in the transportation network. With the transfer arcs $\mathcal{A}_T$ we associate the transfer penalty σ. The lengths τ_p of the passenger paths in the so-constructed passenger routing graph then account for the travel times and transfer penalties. The associated IP model is:

$$
\begin{array}{llll}
\text{(CG)} \quad \min & \lambda \sum_{\ell \in \mathcal{L}} \sum_{f \in \mathcal{F}} c_{\ell,f}\, x_{\ell,f} + (1-\lambda) \sum_{p \in \mathcal{P}} \tau_p\, y_p & & \\
\text{(i)} & \sum_{p \in \mathcal{P}_{st}} y_p & = d_{st} & \forall\, (s,t) \in D \\
\text{(ii)} & \sum_{p : a \in p} y_p & \leq \sum_{\ell : e(a) \in \ell} \sum_{f \in \mathcal{F}} \kappa_{\ell,f}\, x_{\ell,f} & \forall\, a \in \mathcal{A} \\
\text{(iii)} & \sum_{f \in \mathcal{F}} x_{\ell,f} & \leq 1 & \forall\, \ell \in \mathcal{L} \\
\text{(iv)} & \sum_{\ell : e(a) \in \ell} \sum_{f \in \mathcal{F}} f \cdot x_{\ell,f} & \leq F & \forall\, e \in E \\
\text{(v)} & x_{\ell,f} & \in \{0,1\} & \forall\, \ell \in \mathcal{L},\, f \in \mathcal{F} \\
\text{(vi)} & x_{\ell,f} & \geq 0 & \forall\, \ell \in \mathcal{L},\, f \in \mathcal{F} \\
\text{(vii)} & y_p & \geq 0 & \forall\, p \in \mathcal{P}.
\end{array}
$$

The passenger pricing problem is again a shortest path problem, the lines are supposed to be given explicitly.

Proposition 2.4. *Model CG dominates model DCC as an integer program and with respect to the SOS-LP relaxation, i.e.,* $v_{IP}(CG) \geq v_{IP}(DCC)$ *and* $v_{SOS-LP}(CG) \geq v_{SOS-LP}(DCC)$.

Proof. $y_{p,k} = \sum_{P:\psi(P)=p,\, |P \cap \mathcal{A}_T| = k} y_p$ is feasible for (DCC) for any solution of (CG); here $\psi(P)$ denotes for a passenger path P in the change-and-go passenger routing graph $D = (\mathcal{V}, \mathcal{A})$ the corresponding passenger path in the routing graph $\bar{N} = (V, A)$. □

3 Computational Results

We perform in this section a computational comparison of the three line planning models of Section 2, namely, the basic model (B), the direct connection capacity model (DCC), and the change-and-go model (CG). We test the models on six instances that are associated with two underlying transportation networks (three instances for each network) that we denote as `Dutch` and `Potsdam`. In all cases, the line pool can be enumerated; its maximum size is 4,424.

The `Dutch` network was studied by Bussieck [11] as a test case for his direct travelers approach to line planning. Instance `Dutch1` contains 15 lines of a line plan computed by him that maximizes the number of direct travelers. Using the terminal stations of these 15 lines as possible start/end nodes of new lines, we extended the line pool by generating between each pair of terminal nodes lines along shortest paths to obtain instance `Dutch2`; computing all possible lines between each pair of terminals results

Table 1: Statistics on the `Dutch` and the `Potsdam` line planning instances. The columns list the instance, the number of nodes and edges of the preprocessed transportation network, the size of the line pool, the number of nodes and edges of the change-and-go graph, the number of OD-pairs with positive demand, the number of binary variables for lines and frequencies, and the number of constraints for the basic line planning model, the direct connection capacity model, and the change-and-go model.

name	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert D\rvert$	#vars	#constraints (B)	#constraints (DCC)	#constraints (CG)
Dutch1	23	106	15	87	284	420	60	577	737	567
Dutch2	23	106	225	999	5 020	420	900	737	1 462	2 159
Dutch3	23	106	4424	36 326	203 092	420	17 696	5 010	13 358	68 614
Potsdam1	1 089	5 282	51	1 885	8 557	4 443	111	6 966	13 786	6 938
Potsdam2	1 089	5 292	132	3 787	22 914	4 443	342	7 057	18 974	10 485
Potsdam3	1 087	5 268	3433	93 904	702 512	4 443	10 233	10 330	47 155	187 374

in instance `Dutch3`. The line operation frequencies are 3, 6, 9, and 18 (a maximum frequency of 9 is not sufficient to satisfy the entire demand in instance `Dutch1`), and an objective weighing parameter of $\lambda = 0.97$ that produces travel time and cost values of nearly the same order of magnitude. The transfer penalty was set to $\sigma = 15$ minutes.

The `Potsdam` data were provided by the public transport company ViP Verkehrsgesellschaft Potsdam GmbH (ViP) in a joint project to optimize the line plan 2010 for the city of Potsdam. The line pool contains lines for regional and commuter trains. These lines are not operated by ViP and we therefore assume them to be fixed. ViP operates the bus and tram lines in Potsdam, and the task was to plan these lines. The instances `Potsdam1`, `Potsdam2`, and `Potsdam3` result from different stages of the project. Their numbers of edges and nodes differs slightly because of changes in the definition of node and edge attributes throughout the running time of the project, e.g., whether a node is a terminal node or whether turning is possible. In fact, the Potsdam instances are constructed in such a way as to reflect all practical requirements for the line plan 2010, e.g., minimum service frequencies at some stations. Line operation frequencies were taken as 3, 6, and 9 for bus lines; this corresponds to a cycle time of 60, 30, and 20 minutes in a time horizon of 3 hours. Trams had to be operated at frequency 9, i.e., every 20 minutes. Line costs are proportional to line lengths plus a fixed cost term that is used to reduce the number of lines (ViP wanted to operate as few lines as possible). The objective weighing parameter was set to $\lambda = 0.8$, because this value reduces costs significantly while the increase in travel time is small ($\lambda = 0.8$ is an extreme point of the Pareto curve). The transfer penalty was again set to $\sigma = 15$ minutes.

Table 1 gives some statistics about the test instances. The columns labeled $|V|$, $|E|$, and $|\mathcal{L}|$ list the number of nodes, edges, and lines after some preprocessing. We remark that the instances contain copies of edges that account for different transportation modes, e.g., bus, tram, regional, and commuter traffic in Potsdam. The number of nodes and edges of the associated change-and-go graph $|\mathcal{V}|$ and $|\mathcal{E}|$ are listed in the next two columns, followed by the number of OD-pairs $|D|$ with non-zero demand. The last four columns give statistics on the integer programs associated with the three models, namely, the number of binary line/frequency variables (#vars), and the number of constraints in models (B), (DCC), and (CG).

The instances were solved with a column generation algorithm that is implemented on the basis of the CIP framework SCIP, version 1.2.0, see [1, 20], using CPLEX 12.1 [14] as LP-solver (in single core mode). Line/frequency variables were enumerated, passenger path variables were priced using a shortest path algorithm. For the direct connection capacity model we used the pricing algorithm as described in Section 2.2. Some "preprocessing cuts" (about the minimum total line frequency over some cuts, similar to [10]) were added for the Dutch instances, but not for the Potsdam instances because there they took too much time for a small LP improvement. Some cuts of the form

$$\sum_{p\in\mathcal{P}_{st}:a\in p}\frac{y_{p,0}}{d_{st}} \leq \sum_{\ell\in\mathcal{L}_{st}:e(a)\in\ell}\sum_{f\in\mathcal{F}} x_{\ell,f} \qquad \forall\, a \in A,\, \forall\,(s,t)\in D$$

were added to the direct connection capacity models. They stipulate that a direct passenger flow between nodes s and t on some arc a must be covered by at least one line. As these constraints are of the same form as inequalities (DCC) (iii), the pricing problem is essentially unchanged. Furthermore, we used the SOS-constraints instead of SOS branching since the line pool is small enough, and computed integer solutions by means of three special rounding heuristics in addition to the primal heuristics built into SCIP. A time limit was set to 5 hours for the Dutch instances, and to 10 hours for the Potsdam instances, since the Potsdam network is much more complex. All computations were done on an Intel Quad-Core 2, 3.0 GHz computer (in 64 bit mode) with 6 MB cache and 16 GB of main memory, running openSuse Linux 11.2.

Table 2 shows the results. Its columns list the model considered, the solution time (values of 5h and 10h indicate that the time limit has been reached), the number of branch-and-bound nodes, the best lower and upper bounds, and the duality gap. Taking the lines of the best primal solution of a model as line pool of a (small) change-and-go model and solving this model by re-optimizing passenger flow allows to evaluate the quality of line plans that have been computed using models that ignore or underestimate transfer times. The last three columns list the value, the cost and the travel as well as the transfer time of such a "verified" solution.

Only the smallest instance `Dutch1` can be solved to optimality for all models within the given time frame. The only other model that can be solved to optimality is the direct connection capacity model for instance `Potsdam1`. In all other cases, duality gaps come up. No primal solution was found for the largest change-and-go models Dutch3 (CG) and Potsdam3 (CG), and for Dutch3 (CG), the model with the largest number of lines, not even the LP relaxation could be solved.

Comparing different models with respect to verified solutions, it turns out that the

Table 2: Computing line plans with different models. The columns list the instance, computation time, number of branch-and-bound nodes, the best lower and upper bound, the duality gap, and the verified value, cost, and travel time (including transfers) of the best solution.

name	time	nodes	obj.	best sol	gap	sol*	cost*	time*
Dutch1 (B)	1s	111	445 205	445 205	0.00%	461 785	64 900	13 294 394
Dutch1 (DCC)	2s	102	459 493	459 493	0.00%	459 493	66 100	13 179 194
Dutch1 (CG)	13s	1979	459 493	459 493	0.00%	459 493	66 100	13 179 194
Dutch2 (B)	5h	$1.5 \cdot 10^6$	435 341	438 968	0.83%	451 735	58 200	13 176 026
Dutch2 (DCC)	5h	15 989	438 991	452 356	3.04%	452 794	61 600	13 101 414
Dutch2 (CG)	5h	370	4307 100	459 779	6.75%	459 779	76 700	12 845 992
Dutch3 (B)	5h	91 071	434 434	438 742	0.99%	453 699	58 300	13 238 268
Dutch3 (DCC)	5h	138	435 394	491 598	12.91%	490 243	102 000	13 043 446
Dutch3 (CG)	5h	0	-	-	-	-	-	-
Potsdam1 (B)	10h	30 062	201 917	201 961	0.02%	217 187	11 428	1 040 219
Potsdam1 (DCC)	4 522s	69	215 225	215 225	0.00%	215 252	12 565	1 025 999
Potsdam1 (CG)	10h	2 038	214 726	215 468	0.35%	215 468	13 541	1 023 177
Potsdam2 (B)	10h	4 167	200 886	202 174	0.64%	219 822	11 844	1 051 730
Potsdam2 (DCC)	10h	345	213 390	214 783	0.65%	215 086	13 054	1 023 178
Potsdam2 (CG)	10h	121	211 097	215 944	2.30%	215 944	11 625	1 033 220
Potsdam3 (B)	10h	1 821	200 571	202 570	1.00%	220 430	12 580	1 051 823
Potsdam3 (DCC)	10h	1	211 561	217 161	2.65%	217 520	13 054	1 035 382
Potsdam3 (CG)	10h	1	210 445	-	-	-	-	-

change-and-go model is always outperformed by one of the other models within the given time frame. The best solutions for instances `Dutch2` and `Dutch3` were found by the basic model. In all other cases and, in particular, for all Potsdam instances, the direct connection capacity model performs best.

Tables 3 and 4 provide a closer look at the error that is made by ignoring transfer times in the basic model, and by underestimating them in the direct connection capacity model. The tables list intervals of travel times and numbers of transfers, respectively. The following columns give the differences in travel time and transfers, respectively, in number of passengers and in percent, between the original and the verified solution. The basic model performs poorly, while the direct connection capacity model turns out to be amazingly accurate.

References

[1] T. Achterberg, *SCIP: Solving Constraint Integer Programs*, Math. Programming Computation, 1 (2009), pp. 1–41.

Table 3: Comparing original and verified passenger travel times in minutes (including a transfer penalty of 15 minutes) for line plans computed with models (B) and (DCC). The columns list the travel time interval and the differences between the original and the verified solution in number of passengers and in percent for the three `Potsdam` instances.

basic model: original vs. verified solution			
travel time	Potsdam1	Potsdam2	Potsdam3
0 - 10	-74 (0.55%)	-291 (2.2%)	-309 (2.3%)
10 - 20	-2498 (14.1%)	-3020 (17.4%)	-3218 (18.8%)
20 - 40	+1526 (18.4%)	+2334 (28.8%)	+2544 (27.8%)
40 - 60	+650 (9.4%)	+514 (7.6%)	+490 (7.1%)
60 - 90	+367 (28.3%)	+434 (32.5%)	+463 (34.6%)
> 90	+29 (43.9%)	+29 (43.9%)	+30 (45.5%)

direct connection capacity model: original vs. verified solution			
travel time	Potsdam1	Potsdam2	Potsdam3
0 - 10	+29 (0.22%)	+56 (0.42%)	+66 (0.49%)
10 - 20	-29 (0.18%)	-33 (0.18%)	-82 (0.45%)
20 - 40	-6 (0.08%)	-40 (0.53%)	-6 (0.08%)
40 - 60	-2 (0.03%)	-32 (0.47%)	-5 (0.07%)
60 - 90	+8(0.61%)	+49 (3.7%)	+27 (2.0%)
> 90	0 (0%)	0 (0%)	0 (0%)

Table 4: Comparing original and verified passenger transfers for line plans computed with models (B) and (DCC). The columns list the travel time interval and the differences between the original and the verified solution in number of transfers and in percent for the three `Potsdam` instances.

basic model: original vs. verified solution			
#transfers	Potsdam1	Potsdam2	Potsdam3
0	-4131 (11.2%)	-4884 (13.5%)	-4638 (12.7%)
1	+4042 (37.5%)	+4769 (41.6%)	+4490 (40.4%)
2	+89 (65.9%)	+114 (71.3%)	+147 (75.8%)
≥ 3	+1 (inf)	+1 (inf)	+1 (inf)

direct connection capacity model: original vs. verified solution			
#transfers	Potsdam1	Potsdam2	Potsdam3
0	0 (0%)	0 (0%)	0 (0%)
1	-1 (0.01%)	-30 (0.31%)	-28 (0.29%)
2	+1 (0.76%)	+30 (21.7%)	+27 (12.2%)
≥ 3	0 (0%)	0 (0%)	+1 (50%)

[2] E. Beale and J. Tomlin, *Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables*, in Proceedings of the 5th International Operations Research conference, J. Lawrence, ed., Tavistock Publkications Ltd., London, 1970, pp. 447–454.

[3] R. Borndörfer, M. Grötschel, and M. E. Pfetsch, *A column-generation approach to line planning in public transport*, Transportation Science, 1 (2007), pp. 123–132.

[4] ——, *Models for line planning in public transport*, in Computer-aided Systems in Public Transport, M. Hickman, P. Mirchandani, and S. Voß, eds., vol. 600 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 2008, pp. 363–378.

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

[6] M. R. Bussieck, P. Kreuzer, and U. T. Zimmermann, *Optimal lines for railway systems*, Eur. J. Oper. Res., 96 (1997), pp. 54–63.

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

[8] M. R. Bussieck, T. Winter, and U. T. Zimmermann, *Discrete optimization in public rail transport*, Math. Program., 79 (1997), pp. 415–444.

[9] M. T. Claessens, N. M. van Dijk, and P. J. Zwaneveld, *Cost optimal allocation of rail passanger lines*, Eur. J. Oper. Res., 110 (1998), pp. 474–489.

[10] A. Dix, *Das statische Linienplanungsproblem*, diploma thesis, TU Berlin, 2007.

[11] M. B. GAMS, *lop.gms: Line optimization*. http://www.gams.com/modlib/libhtml/lop.htm.

[12] 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.

[13] ——, *A branch-and-cut approach for solving railway line-planning problems*, Transportation Sci., 38 (2004), pp. 379–393.

[14] IBM, *ILOG CPLEX*. http://www-01.ibm.com/software/integration/optimization/cplex/.

[15] K. Nachtigall and K. Jerosch, *Simultaneous network line planning and traffic assignment*, in ATMOS 2008 - 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems, M. Fischetti and P. Widmayer, eds., DROPS, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2008.

[16] 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.

[17] D. M. Ryan and B. A. Foster, *An integer programming approach to scheduling*, in Computer Scheduling of Public Transport, A. Wren, ed., North-Holland Publishing Company, 1981.

[18] A. Schöbel and S. Scholl, *Line planning with minimal traveling time*, in Proc. 5th Workshop on Algorithmic Methods and Models for Optimization of Railways, L. G. Kroon and R. H. Möhring, eds., 2006.

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

[20] *SCIP – Solving Constraint Integer Programs, documentation*. http://scip.zib.de.