Models for Railway Track Allocation*

Ralf Borndörfer and Thomas Schlechte

Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB),
Takustr. 7, 14195 Berlin-Dahlem, Germany,
Email {borndoerfer, schlechte}@zib.de

Abstract. The optimal track allocation problem (OPTRA) is to find, in a given railway network, a conflict free set of train routes of maximum value. We study two types of integer programming formulations for this problem: a standard formulation that models block conflicts in terms of packing constraints, and a novel formulation of the 'extended' type that is based on additional 'configuration' variables. The packing constraints in the standard formulation stem from an interval graph and can therefore be separated in polynomial time. It follows that the LP-relaxation of a strong version of this model, including all clique inequalities from block conflicts, can be solved in polynomial time. We prove that the LP-relaxation of the extended formulation can also be solved in polynomial time, and that it produces the same LP-bound. Albeit the two formulations are in this sense equivalent, the extended formulation has advantages from a computational point of view. It features a constant number of rows and is amenable to standard column generation techniques. Results of an empirical model comparison on mesoscopic data for the Hanover-Fulda-Kassel region of the German long distance railway network involving up to 570 trains are reported.

Key words: track allocation, train timetabling, integer programming, column generation

1 Introduction

Routing trains in a conflict-free way through a network of tracks is one of the basic and at the same time most difficult questions in railway scheduling. The need to coordinate the use of shared infrastructure and the complex operation of this infrastructure using switches and signals impose a great variety of technical constraints, that give rise to a complex problem in which many factors have to be considered simultaneously, see Huisman et al. [2005] and Caprara et al. [2007] for comprehensive surveys.

We consider in this paper the *track allocation problem* to simultaneously determine a set of routes for individual trains through a network. These routes have to be conflict-free in the sense that the headway between two trains on the

*This work was funded by the BMWi, project *Trassenbörse*, grant 19M4031A.

same track must be large enough for safety reasons. Degrees of freedom include the implementation or omission of a route, the choice of a path through the network, and adjustments of departure and arrival times. The goal is to maximize a sum of proceedings associated with each scheduled route. The problem comes up in an auctiong approach to railway track capacity, see Borndörfer et al. [2006].

The track allocation problem is equivalent to the *train timetabling problem*, see Brännlund et al. [1998], Caprara et al. [2001], and Caprara et al. [2002]. The solution of a track allocation problem defines a timetable, which, however, is in general not periodic. This is a big difference to timetabling by *periodic event scheduling*, see the thesis of Liebchen [2006] for an extensive survey.

The track allocation problem is further related to the *train platforming problem*, which also deals with conflict-free routings in stations, but adds parking in sidings, see Kroon et al. [2007]. This problem is usually studied at a much finer level of detail with respect to the infrastructure than the track allocation problem, which is generally considered on macroscopic networks.

Among the earliest theoretical optimization approaches to track allocation problems are integer programming formulations that model train routes as paths in appropriate networks. As early as 1956, Charnes & Miller [1956] propose a set covering formulation, in which 'crew and engine packages' are assigned to circular routes in a railway network; the model is solved with what we would call today a column generation procedure.

Set packing versions of this formulation, which can rule out block conflicts between train routes, have been proposed and studied by a number of authors including Brännlund et al. [1998], Caprara et al. [2001], Caprara et al. [2002], Borndörfer et al. [2006], Cacchiani et al. [2007] and Cacchiani [2007]. The main difficulty with this type of formulation is that it contains a very large number of constraints which makes these models computationally hard, if not intractable, beyond a certain size.

We propose in this article a novel formulation for train routing in an attempt to resolve this difficulty. Our formulation is of the 'extended' type; it rules out conflicts between trains using additional 'configuration' variables. It can be shown that such a model is equivalent to a strong version of the standard packing model (including all clique constraints from conflicts) with respect to both quality and computational complexity of the LP-bound. From a practical point of view, the extended model has the advantage that it is amenable to standard column generation techniques and therefore well suited to solve large-scale problems.

The article is organized as follows. Section 2 gives a formal statement of the optimal track allocation problem. For the sake of clarity of exposition, we concentrate here on a basic version that considers a very simple type of conflicts between trains that we call 'block conflicts'. Packing IP-formulations for the track allocation problem are studied in Section 3.1. We show that block conflicts arise from an interval graph, that cliques from block conflicts can be separated in polynomial time, and that the LP-relaxation of a packing model including all such clique constraints can be solved in polynomial time. Section 3.2 introduces our extended formulation. We show that the pricing problem for configuration variables can be solved by computing a longest path in an appropriately defined acyclic digraph, and that the LP-relaxation of the extended model can also be solved in polynomial time. Section 3.3 compares both models analytically; it turns out that they produce the same LP-bound. The final Section 4 contains a computational model comparison on data for the Hanover-Kassel-Fulda part of the long distance network of the German railway company Deutsche Bahn AG with up to 570 trains.

2 The Optimal Track Allocation Problem

The optimal track allocation problem, also known as the train routing problem or the train timetabling problem, can be formally described as follows. We are given a set I of *requests* to route *trains* in a *train routing digraph* $D = (V, A)$; we allow that D contains multiple arcs between two nodes. D is based on an *infrastructure digraph* $G = (S, J)$, whose nodes and arcs model *stations* and *tracks*, respectively. The train routing digraph is a time expansion of the infrastructure digraph, i.e., the nodes of D model possible *departures* and *arrivals* of trains at stations at certain points in time, the arcs possible timetabled *trips* of specific trains. Formally, we associate with each node $v \in V$ a station $s(v) \in S$ and a discrete time $t(v) \in \mathbb{Z}$. An arc $uv \in A$ models a trip on track $s(u)s(v) \in J$ for a train $i(uv) \in I$, which departs at time $t(u)$ and arrives at time $t(v)$; we assume $t(u) < t(v)$ for all trips $uv \in A$ such that D is acyclic. We associate with train $i \in I$ the trips $A_i := \{a \in A : i(a) = i\} \subseteq A$ that this train can run and the *individual train routing digraph* $D_i := (V, A_i) \subseteq D$, which we assume to contain two special (if need be artificially constructed) nodes s_i and t_i, called *source* and *sink*, that represent the departure and the arrival of train i; we therefore assume $\delta_i^-(s_i) = \delta_i^+(t_i) = \emptyset$ (where $\delta^-(v)$ denotes the set of arcs entering $v \in V$, $\delta^+(v)$ the set of arcs leaving $v \in V$, and $\delta_i^{\pm}(U) := \delta^{\pm}(U) \cap A_i$, $\forall U \subseteq V$), and denote $U_i := V \setminus \{s_i, t_i\}$. A *route* for train i is an s_it_i-path in D_i. Denote the set of all routes for train i by P_i, and the set of all possible routes by P (let P be the disjoint union of the sets P_i, i.e., we distinguish identical routes for different trains). Figure 1 illustrates this construction.

We say that an arc $uv \in A$ occupies or *blocks* its associated track $s(u)s(v)$ for the time interval $[t(u), t(v)-1]$, and that there is a *block conflict* between two arcs u_1v_1 and u_2v_2 on the same track if their track occupation time intervals overlap,

symbol	description	symbol	description
S	stations	$G = (S, J)$	infrastructure digraph
J	tracks	$D = (V, A)$	train routing digraph
I	trains	$D_i = (V, A_i)$	individual routing digraph
w	arc weights	s_i, t_i	source, sink of train i

Table 1: Notation for the optimal track allocation problem (OPTRA).

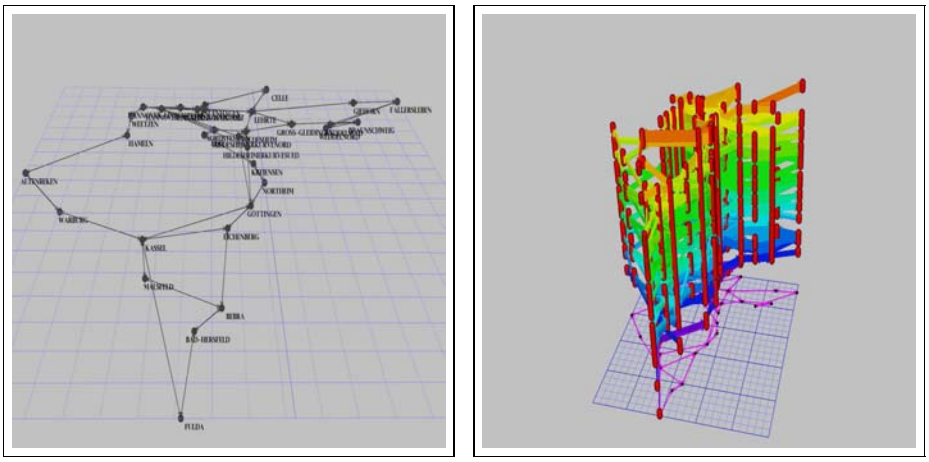

Fig. 1: Infrastructure network (left), and train routing digraph (right); individual train routing digraphs bear different colors.

i.e., if $s(u_1)s(v_1) = s(u_2)s(v_2)$ and $[t(u_1), t(v_1)-1] \cap [t(u_2), t(v_2)-1] \neq \emptyset$. There is a block conflict between two train routes if any of their arcs have a block conflict. A timetable or *schedule* is a set $X \subseteq P$ of conflict-free routes, at most one for each train request, i.e., $|X \cap P_i| \leq 1$, $i \in I$. Assigning weights $w_{uv} \in \mathbb{Z}$ to the arcs $uv \in A$ (modeling 'profits' for individual trips), the weight of route $p \in P$ is $w_p := \sum_{a \in p} w_a$, and the weight of a schedule $X \subseteq P$ is $w(X) := \sum_{p \in X} w_p$. The *optimal track allocation problem* (OPTRA) is to find a schedule of maximum weight.

Caprara et al. [2002] have shown that the stable set problem can be reduced to OPTRA, such that the problem is $\mathcal{NP}$-hard. Indeed, OPTRA can be seen as a problem to find a maximum weight packing (with respect to block conflicts) of train routes in a time-expanded digraph. This framework is fairly general, see the articles of Caprara et al. [2001], Caprara et al. [2002], Cacchiani et al. [2007], Cacchiani [2007] and Borndörfer et al. [2006] for comprehensive discussions how such a model can be used to deal with various kinds of technical constraints.

There is, however, one point where our exposition resorts to a genuine simplification, namely, by considering only block conflicts arising from time overlaps. Such a model obviously ignores important aspects such as different block occupation times for the head and the tail of a train, safety margins to open and close a block after a train has left a track and before it can enter, different driving times of trains (a fast train following a slow train needs a larger safety margin than a slow train following a fast train) etc. Such considerations give rise to headway constraints that guarantee a minimal safety distance in time between two trains on the same track. Such constraints produce more complicated arc conflicts. Namely the ordered pair of arcs u_1v_1 and u_2v_2 on the same track are in conflict, if they fall short of some minimal headway $\tau_{u_1v_1,u_2v_2}$, i.e., $t(u_2) - t(u_1) < \tau_{u_1v_1,u_2v_2}$, see Lukac [2004] for a discussion of such a model involving 'quadrangle-linear headway matrices'. One can show that most of the results of the following sections carry over to more general situations of this type. We do, however, not give the details here, because they would result in a more technical and complicated discussion.

3 Integer Programming Models

3.1 Packing Models

The standard formulation for the track allocation problem models train routes as a multi-commodity flow and rules out block conflicts using additional packing constraints. We need the following additional terminology. Let $B = \{\{a,b\} \in 2^A : a \neq b$ have a block conflict$\}$ be the set of all block conflicts between any two arcs, $H = (A, B)$ the associated (undirected) *(block) conflict graph* (note that the nodes of H are the arcs of the train routing digraph D), and $C = C(H)$ be the set of all (inclusion) maximal *cliques* in H; Figure 2 illustrates the construction of a block conflict graph for a single track.

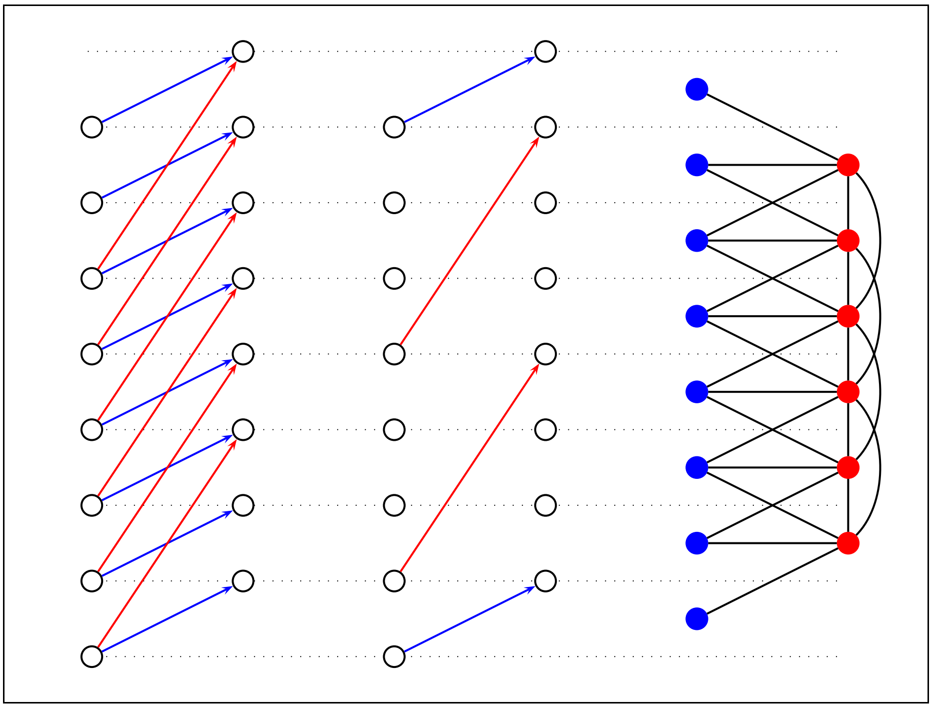

Fig. 2: Block conflicts on a single track: trips for a slow (blue) and a fast (red) train (left), a conflict-free configuration of four trips on this track (middle), and the block conflict graph associated with the track (right).

The packing model comes in two versions, one with 0/1 arc variables x_a, $a \in A$, for the use of trip a in a route, and the other with 0/1 path variables x_p, $p \in P$, for the use of route p. The resulting formulations, we call them *arc packing problem* (APP) and *path packing problem* (PPP), read as follows:

$$
\begin{array}{lll}
\text{(APP)} & \max \sum_{a\in A} w_a x_a & \\
\text{(i)} & \sum_{a\in\delta_i^+(v)} x_a - \sum_{a\in\delta_i^-(v)} x_a = 0 & \forall i \in I, v \in W_i \\
\text{(ii)} & \sum_{a\in\delta_i^+(s_i)} x_a \le 1 & \forall i \in I \\
\text{(iii)} & \sum_{a\in c} x_a \le 1 & \forall c \in C \\
\text{(iv)} & x_a \ge 0 & \forall a \in A \\
\text{(v)} & x_a \in \mathbb{Z} & \forall a \in A
\end{array}
$$

$$
\begin{array}{lll}
\text{(PPP)} & \max \sum_{p\in P} w_p x_p & \\
\text{(ii)} & \sum_{p\in P_i} x_p \le 1 & \forall i \in I \\
\text{(iii)} & \sum_{p\cap c\neq\emptyset} x_p \le 1 & \forall c \in C \\
\text{(iv)} & x_p \ge 0 & \forall p \in P \\
\text{(v)} & x_p \in \mathbb{Z} & \forall p \in P.
\end{array}
$$

Equalities (APP) (i) are *flow conservation constraints*; they route train i on $s_i t_i$-paths; note that D_i is acyclic such that no cycles can come up. Constraints (APP)/(PPP) (ii) ensure a train is routed at most once. The *clique inequalities* (APP)/(PPP) (iii) rule out block conflicts. Finally, (APP)/(PPP) (iv) and (v) are the *nonnegativity* and the *integrality constraints*. Note that all constraints together imply that all variables are 0/1.

The formulations (APP) and (PPP) are strong in the sense that they include all clique constraints from block conflicts. The literature usually considers models that replace (APP)/(PPP) (iii) by weaker constraints

$$
\text{(iii}'\text{)} \quad x_a + x_b \le 1 \quad \forall ab \in B \qquad\qquad \text{(iii}'\text{)} \sum_{p\cap\{a,b\}\neq\emptyset} x_p \le 1 \quad \forall ab \in B
$$

that rule out block conflicts on pairs of arcs; let us denote these variants by (APP′) and (PPP′). Here are some basic properties of the packing models. By definition:

Observation 1 *The block conflict graph $H = (A, B)$ that is associated with an optimal track allocation problem is an interval graph.*

The cliques in the conflict graph are collections of compact real intervals. By Helly's Theorem, see Helly [1923], the intervals of each such clique $c \in C$ contains a common point $t(c)$, and it is easy to see that we can assume $t(c) \in t(V) = \{t(v) : v \in V\}$. It follows that the block conflict graph H has $O(V)$ inclusion maximal cliques, which can be enumerated in polynomial time, and that the packing formulations of the optimal track allocation problem have the sizes listed in Table 2; here, $O(I \times V) + O(I) + O(C) = O(A)$, and we write $O(A) = O(|A|)$ etc.

The LP-relaxation of (APP) can then be solved in polynomial time. To obtain the same result for (PPP), consider a column generation approach. Note

formulation	variables	non-trivial constraints
APP	$O(A)$	$O(A)$
PPP	$O(P)$	$O(V)$
APP ′	$O(A)$	$O(A^2)$
PPP ′	$O(P)$	$O(A^2)$

Table 2: Sizes of packing formulation for the track allocation problem.

that no two arcs in a route are in conflict, i.e., $p \cap c \leq 1$ for all routes $p \in P$ and all cliques $c \in C$. Introducing dual variables γ_i, $i \in I$, for the constraints (PPP) (ii), and η_c, $c \in C$, for the constraints (PPP) (iii), the pricing problem for a route $p \in P_i$, for some train $i \in I$, is

$$\exists\, p \in P_i : \gamma_i + \sum_{p \cap c \neq \emptyset} \eta_c < w_p \iff \sum_{a \in p} (w_a - \sum_{c \ni a} \eta_c) > \gamma_i.$$

This is a longest $s_i t_i$-path problem in the acyclic digraph $D_i = (V, A_i)$ w.r.t. arc weights $w_a - \sum_{a \in c} \eta_c$; this problem can be solved in polynomial time (in fact, in linear time). By the polynomial equivalence of separation and optimization, see Grötschel et al. [1988], here applied to the dual of (PPP), i.e., the polynomial equivalence of pricing and optimization, we obtain the desired result.

Theorem 2. *The LP-relaxations associated with the strong arc packing formulation* APP *and the strong path packing formulation* PPP *of the optimal track allocation problem can be solved in polynomial time.*

3.2 Extended Models

We propose in this section an alternative formulation for the optimal track allocation problem that guarantees a conflict free routing by allowing only feasible route combinations, and not by excluding conflicts. The formulation is based on the concept of feasible arc *configurations*, i.e., sets of arcs on a track without block conflicts. Formally, we define a configuration for some track $j = xy \in J$ as a set of arcs $q \subseteq A_j := \{uv \in A : s(u)s(v) = xy\}$ such that

$$|q \cap c| \leq 1 \quad \forall c \in C.$$

Denote by Q_j the set of all such configurations for track j, $j \in J$, and by Q the set of all such configurations. The idea of the extended model is to introduce 0/1 variables y_q for choosing a configuration on each track and to force a conflict free routing of trains through these configurations by means of inequalities

$$\sum_{p \ni a} x_p \leq \sum_{q \ni a} y_q \quad \forall a \in A.$$

Instead of directly writing down a corresponding model, however, we propose a version that will model configurations as paths in a certain acyclic routing digraph. The advantages of such a formulation will become clear in a minute. The construction extends the routing digraph $D = (V, A)$ to a larger digraph $\overline{D} = (\overline{V}, \overline{A})$ by adding nodes and arcs as illustrated in Figure 3. The details are as follows. Consider a track $xy \in J$ and the trips $A_{xy} = \{uv \in A : s(u)s(v) = xy\}$

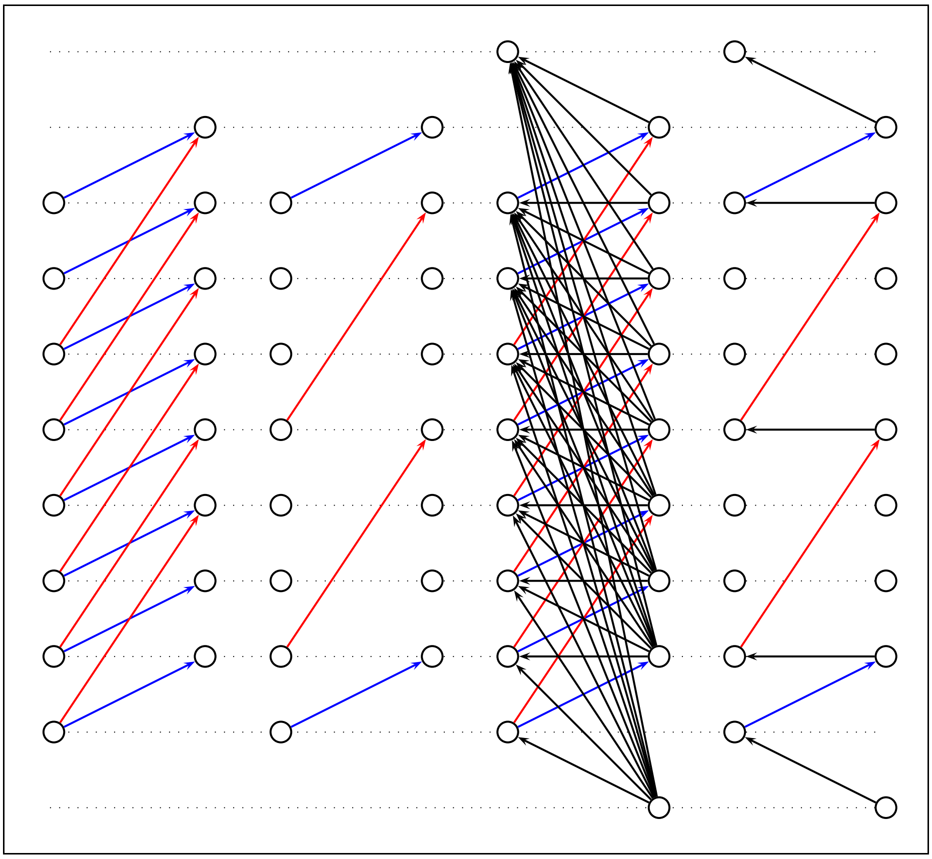

Fig. 3: Configuration routing digraph for a single track: train routing digraph (left), configuration (half-left), configuration routing digraph (half-right), and the corresponding path (right).

on this track. Denote by $L_{xy} := \{u : uv \in A_{xy}\}$ and $R_{xy} := \{v : uv \in A_{xy}\}$ the associated set of departure and arrival nodes. Construct two new, additional nodes s_{xy} and t_{xy} by setting $s(s_{xy}) = y$, $t(s_{xy}) := \min t(R_{xy}) - 1$, and $s(t_{xy}) = x$, $t(t_{xy}) := \max t(R_{xy}) + 1$, i.e., s_{xy} marks an artificial source node at station y before the departure of the earliest trip on xy, and t_{xy} marks an artificial sink node at station x after the arrival of the latest trip on xy. Let $\overline{L}_{xy} := L_{xy} \cup \{t_{xy}\}$ and $\overline{R}_{xy} := R_{xy} \cup \{s_{xy}\}$; note that all arcs in A_{xy} go from $\overline{L}_{xy}$ to $\overline{R}_{xy}$ (actually from L_{xy} to R_{xy}). Now let $\overline{A}_{xy} := \{vu : t(v) \leq t(u), v \in \overline{R}_{st}, u \in \overline{L}_{st}\}$ be a set of 'return' arcs that go in the opposite direction; they connect the arrival of a trip on xy (or node s_{xy}) with all possible follow-on trips (or node t_{xy}) on that track. It is easy to see that the *configuration routing digraph* $\overline{D}_{xy} := (\overline{L}_{xy} \cup \overline{R}_{xy}, A_{xy} \cup \overline{A}_{xy})$ is bipartite and acyclic, and that $s_{xy}t_{xy}$-paths $a_1, \overline{a}_1, \ldots, \overline{a}_{k-1}, a_k$ in $\overline{D}_{xy}$ and configurations $a_1, \ldots, a_k$ in Q_{st} are in 1-1 correspondence. Let us formally denote this isomorphism by a mapping

$$\bar{\cdot} : Q_j \to \overline{Q}_j, \quad q \mapsto \overline{q}, \qquad j \in J,$$

where $\overline{Q}_j$ denotes the set of all $s_j t_j$-paths in $\overline{D}_j$; however, we will henceforth identify paths $\overline{q} \in \overline{Q}_j$ and configurations $q \in Q_j$. Let us also denote by $U_j := L_j \cup R_j$ the structural nodes of $\overline{D}_j$, and by $\overline{D} := (\overline{V}, \overline{A}) := (V \cup \{s_j, t_j : j \in J\}, A \cup \bigcup_{j \in J} \overline{A}_j) = \bigcup_{j \in J} \overline{D}_j$ the *extended train routing digraph*, i.e., the routing digraph D extended by the artificial nodes and return arcs described above, and $\delta_j^{\pm}(W) := \delta^{\pm}(W) \cap A_j \cup \overline{A}_j$, $\forall W \subseteq \overline{V}$.

The extended model also comes in two versions, one using new 0/1 arc variables y_a, $a \in \overline{A}$, for the use of arc a in a configuration-path, and the other with 0/1 path variables y_q, $q \in Q$, for the use of configuration-path $q \in Q$. The resulting formulations, which we call *arc configuration problem* (ACP) and *path configuration problem* (PCP), read as follows:

(ACP)			(PCP)		
	$\max \sum_{a \in A} w_a x_a$			$\max \sum_{p \in P} w_p x_p$	
(i)	$\sum_{a \in \delta_i^+(v)} x_a - \sum_{a \in \delta_i^-(v)} x_a = 0$	$\forall i \in I, v \in W_i$			
(ii)	$\sum_{a \in \delta_i^+(s_i)} x_a \le 1$	$\forall i \in I$	(ii)	$\sum_{p \in P_i} x_p \le 1$	$\forall i \in I$
(iii)	$\sum_{a \in \delta_j^+(v)} y_a - \sum_{a \in \delta_j^-(v)} y_a = 0$	$\forall j \in J, v \in U_j$			
(iv)	$\sum_{a \in \delta_j^+(s_j)} y_a \le 1$	$\forall j \in J$	(iv)	$\sum_{q \in Q_j} y_q \le 1$	$\forall j \in J$
(v)	$x_a - y_a \le 0$	$\forall a \in A$	(v)	$\sum_{p \ni a} x_p - \sum_{q \ni a} y_q \le 0$	$\forall a \in A$
(vi)	$x_a \ge 0$	$\forall a \in A$	(vi)	$x_p \ge 0$	$\forall p \in P$
(vii)	$y_a \ge 0$	$\forall a \in A$	(vii)	$y_q \ge 0$	$\forall q \in Q$
(viii)	$x_a \in \mathbb{Z}$	$\forall a \in A$	(viii)	$x_p \in \mathbb{Z}$	$\forall p \in P$
(ix)	$y_a \in \mathbb{Z}$	$\forall a \in A$	(ix)	$y_q \in \mathbb{Z}$	$\forall q \in Q.$

Equalities (ACP) (i) and (iii) are *flow conservation constraints*; they route trains i on $s_i t_i$-paths and configurations j on $s_j t_j$-paths; note that both D_i and $\overline{D}_j$ are acyclic such that no cycles can come up. Constraints (ACP)/(PCP) (ii) and (iv) ensure a train is routed at most once and that at most one configuration can be chosen for each track. The *coupling constraints* (ACP)/(PCP) (v) synchronize routes and configurations. Finally, (APP)/(PPP) (iv) and (v) are the *nonnegativity* and the *integrality constraints*. Note that, again, all variables are implicitly 0/1.

formulation	variables	non-trivial constraints
ACP	$O(A)$	$O(A)$
PCP	$O(P)+O(Q)$	$O(I)+O(J)$

Table 3: Sizes of packing formulation for the track allocation problem.

The extended models have the sizes listed in Table 3. Then the LP-relaxation of (ACP) can be solved in polynomial time. For (PCP), consider the pricing problems for routes and configurations. With dual variables γ_i, $i \in I$, π_j, $j \in J$, and λ_a, $a \in A$, for constraints (PCP) (ii), (iv), and (v), respectively, the pricing problem for a route $p \in P_i$ for train $i \in I$ is

$$\exists\, p \in P_i : \gamma_i + \sum_{a \in p} \lambda_a < w_p \iff \sum_{a \in p} (w_a - \lambda_a) > \gamma_i.$$

This is the same as finding a longest $s_i t_i$-path in D_i w.r.t. arc weights $w_a - \lambda_a$; as D_i is acyclic, this problem can be solved in polynomial time. The pricing problem for a configuration $q \in Q_j$ for track $j \in J$ is

$$\exists\, q \in Q_j : \pi_j - \sum_{a \in q} \lambda_a < 0 \iff \sum_{a \in q} \lambda_a > \pi_j.$$

Using arc weights λ_a, $a \in A_j$, and 0, $a \in \overline{A}_j$, pricing configurations in Q_j is the same as finding longest $s_j t_j$-paths in the acyclic digraph $\overline{D}_j$. This is polynomial. We conclude:

Theorem 3. *The LP-relaxations associated with the arc configuration formulation* ACP *and the path configuration formulation* PCP *of the optimal track allocation problem can be solved in polynomial time.*

Let us quickly state in this pricing context a simple bound on the LP-value of the path configuration formulation PCP that is useful in practice to overcome tailing-off effects in a column generation procedure. Namely, computing the path lengths $\max_{p \in P_i} \sum_{a \in p} (w_a - \lambda_a)$ and $\max_{q \in Q_j} \sum_{a \in q} \lambda_a$ yield the following LP-bound $\beta = \beta(\gamma, \pi, \lambda)$.

Lemma 1. *Let* $\gamma, \pi, \lambda \geq 0$ *be dual variables*[1] *for* PCP *and* $v_{\text{LP}}(\text{PCP})$ *the optimum objective value of the LP-relaxation of* PCP. *Define*

$$\begin{aligned} \eta_i &:= \max_{p \in P_i} \sum_{a \in p} (w_a - \lambda_a) - \gamma_i, && \forall i \in I, \\ \theta_j &:= \max_{q \in Q_j} \sum_{a \in q} \lambda_a - \pi_j, && \forall j \in J, \\ \beta(\gamma, \pi, \lambda) &:= \sum_{i \in I} \max\{\gamma_i + \eta_i, 0\} + \sum_{j \in J} \max\{\pi_j + \theta_j, 0\}. \end{aligned}$$

[1] Note that these will be infeasible during a column generation.

Then

$$v_{\mathrm{LP}}(\mathrm{PCP}) \leq \beta(\gamma, \pi, \lambda).$$

Proof.

- $\gamma_i + \eta_i \geq \sum_{a \in p} (w_a - \lambda_a) \Rightarrow \gamma_i + \eta_i + \sum_{a \in p} \lambda_a \geq w_p \quad \forall i \in I, p \in P_i.$
- $\pi_j + \theta_j \geq \sum_{a \in q} \lambda_a \Rightarrow \pi_j + \theta_j - \sum_{a \in q} \lambda_a \geq 0 \quad \forall j \in J, q \in Q_j.$
- $(\max\{\gamma + \eta, 0\}, \max\{\pi + \theta, 0\}, \lambda)$ (the maximum taken component-wise) is dual feasible for the LP-relaxation of PCP.

3.3 Model Comparison

We finally compare the two types of models that we have stated. Starting points are the LP-relaxations of the configuration formulations and those of the packing formulations. As the LP-relaxations of APP and PPP, and of ACP and PCP are obviously equivalent via flow decomposition into paths, it suffices to compare, say, APP and ACP.

Lemma 2. *Let*

$$\begin{aligned} P_{\mathrm{LP}}(\mathrm{APP}) &:= \{x \in \mathbb{R}^A : (\mathrm{APP})\ (\mathrm{i})\text{–}(\mathrm{iv})\} \\ P_{\mathrm{LP}}(\mathrm{ACP}) &:= \{(x, y) \in \mathbb{R}^{A \times \overline{A}} : (\mathrm{ACP})\ (\mathrm{i})\text{–}(\mathrm{vii})\} \\ \pi_x &: \mathbb{R}^{A \times \overline{A}} \to \mathbb{R}^A, \quad (x, y) \mapsto x \end{aligned}$$

be the polyhedra associated with the LP-relaxations of APP *and* ACP*, respectively, and a mapping that produces a projection onto the coordinates of the train routing variables. Then*

$$\pi(P_{\mathrm{LP}}(\mathrm{ACP})) = P_{\mathrm{LP}}(\mathrm{APP}).$$

Proof. Let $C_j := \{c \in C : c \subseteq A_j\}$, $j \in J$, be the set of block conflict cliques associated with track j. Consider the polyhedra

$$\begin{aligned} P &:= \{x \in \mathbb{R}^A : (\mathrm{APP})\ (\mathrm{i}),\ (\mathrm{ii}),\ (\mathrm{vi})\}, \\ P^j &:= \{x \in \mathbb{R}_+^{A_j} : \sum_{a \in c} x_a \leq 1 \quad \forall c \in C_j\}, \quad j \in J, \\ Q^j &:= \{y \in \mathbb{R}_+^{A_j \times \overline{A}_j} : \sum_{a \in \delta_j^+(v)} y_a = \sum_{a \in \delta_j^-(v)} y_a, \forall v \in U_j, \sum_{a \in \delta_j^+(s_j)} y_a \leq 1\}, \quad j \in J, \\ R^j &:= \{x \in \mathbb{R}_+^{A_j} : \exists y \in Q^j : x \leq y\}, \quad j \in J. \end{aligned}$$

P^j is integer, because C_j is the family of all maximal cliques of an interval graph, which is perfect; Q^j is integer, because it is the path polytope associated with an acyclic digraph; finally, R^j is integer, because it is the anti-dominant of an integer polytope. Consider integer points, it is easy to see that P^j and R^j coincide, i.e., $P^j = R^j$, $j \in J$. It follows

$$P_{\text{LP}}(\text{APP}) = P \cap \bigcap_{j \in J} P^j = P \cap \bigcap_{j \in J} R^j = \pi(P_{\text{LP}}(\text{ACP})).$$

This immediately implies our main Theorem.

Theorem 4. *Denote by $v(P)$ and $v_{\text{LP}}(P)$ the optimal value of problem P and its LP-relaxation, respectively, $P \in \{\text{APP}, \text{PPP}, \text{ACP}, \text{PCP}\}$. Then:*

- $v_{\text{LP}}(\text{APP}) = v_{\text{LP}}(\text{PPP}) = v_{\text{LP}}(\text{ACP}) = v_{\text{LP}}(\text{PCP})$.
- $v(\text{APP}) = v(\text{PPP}) = v(\text{ACP}) = v(\text{PCP})$.

4 Computational Results

We have implemented model generators for the static formulations APP′ and ACP, and a column generation algorithm for model PCP. This choice is motivated as follows. APP′ is the dominant model in the literature, which we want to benchmark. APP and ACP are equivalent models that improve APP′, both arc-based. ACP is easy to implement. We didn't implement the strong packing model APP, and also not PPP, because these models are not robust w.r.t. changes in the problem structure, namely, their simplicity depends on the particular clique structure of interval graphs. If more complex constraints are considered, these models can become hard to adapt. In fact, the instances that we are going to consider involve headway matrices that give rise to more numerous and more complex clique structures, such that an implementation of suitably extended models APP and PPP would have been much more difficult than an implementation of the basic versions that we have considered in the theoretical part of this paper. On the other hand, headway constraints are easy to implement in a configuration model, because they specify possible follow-on trips on a track, which is precisely what a configuration does. Formulation PCP is in this sense robust. It is also well suited for column generation to deal with large instances. In our experiments, we consider the Hanover-Kassel-Fulda area of the German long-distance railway network. All our instances are based on the mesoscopic infrastructure network that is illustrated in Figure 1. It includes data for 37 stations, 120 tracks and 6 different train types (ICE, IC, RE, RB, S, ICG). Because of various possible turnover and driving times for each train type, this produces an infrastructure digraph with 146 nodes, 1480 arcs, and 4320 headway constraints.

Based on the 2002 timetable of Deutsche Bahn AG, we constructed three scenarios that we denote by 146, 285, and 570. The name of the instance gives the number of train requests, which consist of long distance trains (IC, ICE), synchronized regional and suburban passenger trains (S, RE, RB), and freight trains (ICG). The main objective is to maximize the total number of trains in the schedule; on a secondary level, we slightly penalize deviations from certain desired departure and arrival times. Flexibility to reroute trains is controlled by departure and arrival time windows of length at most τ, where τ is a parameter. Increasing τ from 0 to 30 minutes in steps of 2 minutes increases flexibility, but also produces larger train routing digraphs and IPs. After some preprocessing (eliminating arcs and nodes which cannot be part of a feasible train route), the resulting 48 instances have the sizes listed in Table 4. In this table, column τ gives the length of the departure and arrival time, columns *#nodes* and *#arcs* give the sizes $|V|$ and $|A|$ of the preprocessed train routing digraph D associated with the respective instance.

These 48 instances were solved as follows. The root LP-relaxations of the static models APP′ and ACP were solved with the dual simplex method of CPLEX 10.0, see CPLEX [2006]. Then, CPLEXMIP was called for a maximum of at most 1h of running time or 10.000 nodes[2]. Model PCP is solved by column generation, with a limit of at most 100 iterations. The reduced master-LPs were solved with the barrier or the dual simplex method of CPLEX 10.0, depending on the column generation progress. Then, a heuristic integer solution is constructed, namely, by simply computing an optimal integer solution to the last reduced master-LP, again using CPLEXMIP. All computations were made single threaded on a Dell Precision 650 PC with 2GB of main memory and a dual Intel Xeon 3.8 GHz CPU running SUSE Linux 10.1.

Figures 4, 5, and 6 summarize our results on the three scenarios 146, 285, and 570, increasing the flexibility from 0 to 30 minutes per train in steps of 2 minutes. It turns out that, in fact, model APP′ produces a noticeably weaker LP-bound (upper bound) than the bounds from the other two models, which are more or less identical. This shows that it is possible to solve the LP-relaxation of model PCP by column generation almost to proven optimality. Figure 7 provides a closer look at the master-LP associated with model PCP. Indeed, the upper bound $\beta(\gamma, \pi, \lambda)$ and the value $v(\text{RPLP})$ of the reduced master-LP converge in the column generation process.

With increasing flexibility the models become larger, and at some point the LPs could not be solved any more, because we ran out of memory; the vertical bars in Figures 4, 5, and 6 indicate the largest scenarios that could be solved. $O(A^2)$ constraints kill model APP′ early. Model ACP reaches somewhat farther. However, the dynamic model PCP is the one that is able to solve the largest scenarios. It is, in our opinion, also the model that offers the biggest potential for further algorithmic improvements to deal with even larger instances; we are currently working in this direction.

The best integral solutions for our instances were always provided by model ACP. This is no surprise, because this model outperforms APP′ in terms of the LP-bound, while the simple IP heuristic that we have applied to PCP is obviously improvable. Tables 5 and 6 list the details for the largest scenario 570 for models APP′ and ACP. In addition to the size of the respective LPs we

[2]That means that we do not always report optimal integer solutions; however, we remark that all instances of scenario 146, of scenario 285 up to $\tau = 24$, and of scenario 570 up to $\tau = 4$ can be solved to proven optimality by running CPLEX long enough.

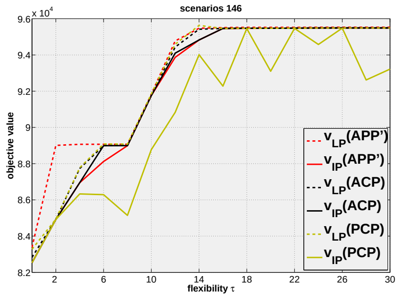

Fig. 4: Solving scenario 146 with models APP′, ACP, and PCP.

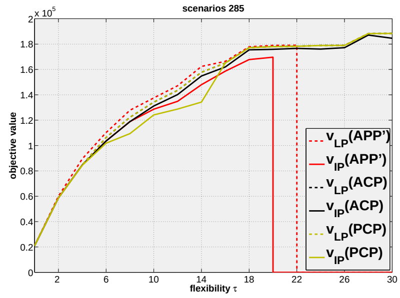

Fig. 5: Solving scenario 285 with models APP′, ACP, and PCP.

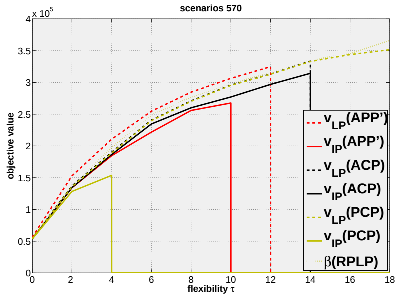

Fig. 6: Solving scenario 570 with models APP′, ACP, and PCP.

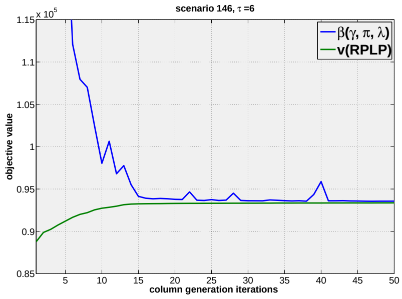

Fig. 7: Generating columns in model PCP for scenario 146.

report the LP and IP values, the overall time $t_{\sum}$, and the time t_{IP} spent on finding integral solutions, both in seconds. The dashes in the tables indicate the inability to compute a solution due to an out of memory error. Table 7 gives similar results for model PCP. Here, the LP sizes refer to the final restricted master-LP, and instead of LP and IP values, we list the lower and upper LP-bounds $v(\text{RPLP})$; instead of IP time, we give the number $\#CG_{iter}$ of column generation iterations. Again, the dashes in the tables report out of memory errors. Altogether, Tables 5, 6, and 7 give an impression of the current performance and the limits of our implementations.

Acknowledgement. We thank three anonymous referees for their valuable comments that helped to improve the quality and the exposition of the material in this article.

Table 4: Test scenarios.

	146		285		570	
τ	*#nodes*	*#arcs*	*#nodes*	*#arcs*	*#nodes*	*#arcs*
0	2877	3297	362	422	1284	1412
2	4953	6414	1501	1846	5858	6894
4	7428	10131	3262	4284	10912	13334
6	9766	13673	5243	7140	19484	25220
8	12143	17300	8070	11289	28038	37128
10	15617	22476	11126	15840	38380	51944
12	19574	28632	15226	22014	50768	70160
14	24142	35886	19970	29325	65056	91648
16	28877	43673	26201	38985	80376	115212
18	33694	51799	32599	49137	97954	142780
20	38953	60707	39854	60920	116886	173516
22	44072	69636	47486	73473	138512	209040
24	50287	80556	56502	88475	161590	247072
26	56156	91019	65579	103979	186458	289266
28	62035	101581	75820	121840	212722	334878
30	69813	115838	87883	143374	241224	383914

Table 5: Solving model APP′ for scenario 570.

τ	*#rows*	*#cols*	v_{LP}	v_{IP}	$t_{\sum}$	t_{IP}
0	1441	1412	56264.17	53676.00	290.27	0.10
2	8760	6894	152778.29	134190.00	400.88	19.97
4	19369	13334	210479.74	184636.00	658.59	42.14
6	44272	25220	254676.53	221725.00	401.15	103.54
8	81313	37128	284689.94	255870.00	538.52	213.84
10	143917	51944	306437.88	267569.00	1210.23	415.15
12	252530	70160	324781.31	-	1761.22	1360.30
14	413828	91648	-	-	-	-
16	637237	115212	-	-	-	-
18	965427	142780	-	-	-	-
20	1436049	173516	-	-	-	-
22	2094272	209040	-	-	-	-
24	2895176	247072	-	-	-	-
26	3999163	289266	-	-	-	-
28	5422512	334878	-	-	-	-
30	7048470	383914	-	-	-	-

Table 6: Solving model ACP for scenario 570.

τ	$\#rows$	$\#cols$	v_{LP}	v_{IP}	$t_{\sum}$	t_{IP}
0	2332	3875	53968.00	53676.00	216.51	0.21
2	11106	19926	136944.50	134311.00	540.97	6.44
4	21772	39967	189997.08	186467.00	622.68	22.60
6	41498	79234	240622.38	234535.00	1495.82	931.92
8	60390	120957	270900.38	260063.00	2170.88	1401.25
10	83398	170277	295798.29	277073.00	4203.54	3488.38
12	111270	231613	313179.33	296917.00	4760.91	3819.11
14	143270	303302	333515.08	314348.00	4361.18	3943.13
16	177622	377312	-	-	-	-
18	215888	461844	-	-	-	-
20	257378	549535	-	-	-	-
22	304326	649176	-	-	-	-
24	354762	754888	-	-	-	-
26	409556	869796	-	-	-	-
28	467950	985555	-	-	-	-
30	529518	1107237	-	-	-	-

Table 7: Solving model PCP for scenario 570.

τ	$\#rows$	$\#cols$	β	v(RPLP)	gap in %	$t_{\sum}$	$\#CG_{iter}$
0	1248	11715	54727.00	53767.00	1.78	468.11	51
2	3314	66012	137376.07	135729.48	1.21	5883.12	100
4	6160	166133	197333.08	188757.73	4.54	13687.55	100
6	11300	238837	248480.85	239768.92	3.63	28258.23	82
8	16414	272565	276867.11	270234.28	2.45	43199.62	82
10	22846	168492	299070.52	295415.44	1.24	73891.21	100
12	30770	214259	314654.48	312960.40	0.54	183123.49	100
14	40696	355918	335061.01	332970.27	0.63	336374.07	57
16	51562	346564	345445.44	343802.93	0.48	198590.48	100
18	63998	266214	366323.70	351502.63	4.22	463379.15	46
20	78478	-	-	-	-	-	
22	94994	-	-	-	-	-	
24	112816	-	-	-	-	-	
26	132826	-	-	-	-	-	
28	154706	-	-	-	-	-	
30	177914	-	-	-	-	-	

Bibliography

Borndörfer, Grötschel, Lukac, Mitusch, Schlechte, Schultz & Tanner (2006). An Auctioning Approach to Railway Slot Allocation. *Competition and Regulation in Network Industries 1(2), 163–196.*

Brännlund, Lindberg, Nou & Nilsson (1998). Railway Timetabling using Langangian Relaxation. *Transportation Science 32(4), 358–369.*

Cacchiani (2007). *Models and Algorithms for Combinatorial Optimization Problems arising in Railway Applications.* PhD thesis, DEIS, Bologna.

Cacchiani, Caprara & Toth (2007). *A Column Generation Approach to Train-Timetabling on a Corridor4OR* . To appear.

Caprara, Fischetti, Guida, Monaci, Sacco & Toth (2001). Solution of Real-World Train Timetabling Problems. In *HICSS 34*. IEEE Computer Society Press.

Caprara, Fischetti & Toth (2002). Modeling and Solving the Train Timetabling Problem. *Operations Research 50(5), 851–861.*

Caprara, Kroon, Monaci, Peeters & Toth (2007). Passenger Railway Optimization. In C. Barnhart & G. Laporte (Eds.), *Handbooks in Operations Research and Management Science*, volume 14 chapter 3, pp. 129–187. Elsevier.

Charnes & Miller (1956). A Model for the Optimal Programming of Railway Freight Train Movements. *Management Science 3(1), 74–92.*

CPLEX (2006). *User-Manual CPLEX 10.0.* ILOG CPLEX Division.

Grötschel, Lovász & Schrijver (1988). *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics.* Springer.

Helly (1923). Über Mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresber. Deutsch. Math. Verein. 32, 175–176.*

Huisman, Kroon, Lentink & Vromans (2005). Operations research in passenger railway transportation. Technical Report EI2005-16, Econometric Institute, Erasmus University Rotterdam.

Kroon, Lentink & Schrijver (2007). Shunting of Passenger Train Units: An Integrated Approach. Technical Report ERIM ERS-2006-068-LIS, Erasmus University Rotterdam.

Liebchen (2006). *Periodic Timetable Optimization in Public Transport.* PhD thesis, Technical University Berlin.

Lukac (2004). Holes, Antiholes and Maximal Cliques in a Railway Model for a Single Track. Technical Report ZIB Report 04-18, Zuse-Institut Berlin, Takustr. 7, 14195 Berlin.