Integrating line planning, timetabling, and vehicle scheduling

Integer programming formulation and analysis*

Marco Lübbecke Christian Puchert Philine Schiewe
Anita Schöbel

February 27, 2019

Abstract

Line planning, timetabling, and vehicle scheduling are three important stages of public transport planning which highly depend on one another. It is hence beneficial to solve them in an integrated way instead of sequentially. In this paper we present an integer linear programming formulation for the integrated line planning, timetabling, and vehicle scheduling problem. Due to the inherent complexity of line planning and timetabling, it is not possible to solve the integrated model directly. We hence analyze the structure of the resulting integer program with respect to decomposition approaches and show that there exist decompositions that are superior to the canonical decomposition into the planning stages line planning, timetabling, and vehicle scheduling.

Keywords Line planning - Timetabling - Vehicle scheduling - Integrated public transport planning - Integer programming - Decomposition

*This work was partially funded by DFG research unit FOR 2083.

M. Lübbecke, C. Puchert
RWTH Aachen University
Kackertstraße 7
52072 Aachen, Germany
e-mail: luebbecke@or.rwth-aachen.de, puchert@or.rwth-aachen.de

P. Schiewe
University of Goettingen
Lotzestr. 16-18
37083 Göttingen, Germany
e-mail: p.schiewe@math.uni-goettingen.de

A. Schöbel
Technical University of Kaiserslautern
Gottlieb-Daimler-Straße, 67663 Kaiserslautern, Germany
e-mail: schoebel@mathematik.uni-kl.de
and
Fraunhofer Institute for Industrial Mathematics ITWM
Fraunhofer Platz 1
67663 Kaiserslautern, Germany

1 Introduction

When planning public transport, three important and well researched stages are line planning, timetabling, and vehicle scheduling. They are usually solved sequentially, see, e.g. (Bussieck et al, 1997; Huisman et al, 2005; Desaulniers and Hickman, 2007; Guihaire and Hao, 2008). Thus, at first a line plan is constructed that covers the infrastructure network. Afterwards, arrival and departure times for all lines and all stops are determined in the timetabling stage minimizing the travel time of the passengers. For the lines and the corresponding timetable a vehicle schedule is constructed such that the operational costs are minimized. Due to the nature of the sequential planning process, the solution quality and even the feasibility of the later problems highly depend on the solution of the earlier problems and the quality of the resulting public transport supply is very likely to benefit from an integrated approach. Already in (Bussieck et al, 1997) it is therefore suggested to consider integrated problems spanning more than one planning stage.

Recently, integration has been the focus of many publications in public transport planning. The integration of timetabling and passenger routing is considered in (Siebert and Goerigk, 2013; Schmidt, 2014; Schmidt and Schöbel, 2015a,b; Borndörfer et al, 2016; Gattermann et al, 2016; Robenek et al, 2017) while line planning and timetabling are considered integratedly in (Schmidt, 2005; Liebchen, 2008; Rittner and Nachtigall, 2009; Kaspi and Raviv, 2013; Burggraeve et al, 2017; Schiewe, 2018). Integrating timetabling and vehicle scheduling is considered in (van den Heuvel et al, 2008; Guihaire and Hao, 2010; Ibarra-Rojas and Rios-Solis, 2011; Cadarso and Marín, 2012; Petersen et al, 2013; Schmid and Ehmke, 2015; Yue et al, 2017; Fonseca et al, 2018). While the integration of several planning stages leads to better solutions, it also increases the computational challenge. Therefore, many approaches to integrating problems in public transport planning are heuristic, such as iterative approaches in (Kinder, 2008; Siebert and Goerigk, 2013; Burggraeve et al, 2017) or meta-heuristic ones in (van den Heuvel et al, 2008; Guihaire and Hao, 2010; Petersen et al, 2013; Schmid and Ehmke, 2015; Robenek et al, 2017; Yue et al, 2017; Fonseca et al, 2018). A general scheme for designing iterative algorithms for integrating line planning, timetabling, and vehicle scheduling is proposed in (Schöbel, 2017), of which parts are implemented in (Michaelis and Schöbel, 2009; Pätzold et al, 2017; Schiewe and Schiewe, 2018).

While the integration of two planning stages is subject of ongoing research, the integration of all three problems line planning, timetabling, and vehicle scheduling has not been handled before. We propose an integrated formulation for all three problems which also includes passenger routing in Section 2 of this paper and consider different decompositions in regard of their solvability in Section 4.

2 Integrating line planning, timetabling, and vehicle scheduling with passenger routing

In this section we shortly describe the usual sequential approach to public transport planning and introduce an IP model for the integrated problem.

2.1 Sequential solution

While the sequential planning stages are thoroughly researched, we only give a brief introduction to each of them. Note that there are many different models for each of the planning stages which are described in the overview articles referenced below.

Line planning. The goal of line planning is to cover an infrastructure network by lines such that a certain travel quality for the passengers is guaranteed and the costs for the infrastructure provider is not too high. This is usually done by routing all passengers along shortest paths and assigning lower and upper frequency bounds on the edges of the infrastructure network that make sure that these passengers paths can be realized. For an overview, see (Schöbel, 2012).

Timetabling. We consider periodic timetabling, i.e., the timetable is constructed for a fixed period and then repeated. This is usually modeled as periodic event scheduling problem (PESP), as introduced in (Serafini and Ukovich, 1989), which is intrinsically hard to solve, see e.g. (Liebchen, 2007). To avoid integrating passenger routing, it is often assumed that passengers travel on fixed paths independently from the timetable. For an overview, see (Lusby et al, 2011).

Vehicle scheduling. In comparison to line planning and timetabling, vehicle scheduling is an easier problem as its basic form can be solved by a flow formulation. The goal is to minimize the operational costs by scheduling the operation of lines by vehicles such that additional costs arising from empty relocation trips between line trips are minimized. It is an aperiodic problem as vehicle schedules do not need to repeat with the same period as the timetable does. For an overview, see (Bunte and Kliewer, 2009).

Passenger routing. Passenger routing is used in different stages of the sequential approach. It is used to find lower frequency bounds in the line planning stage as well as for defining weights in the timetabling stage. In the objective function it is used to evaluate timetables with respect to the passengers' travel times. We have to consider passenger routing in the integrated model, because assigning passengers to transfers beforehand is impossible when no line plan is fixed.

2.2 Integer programming (IP) model

Our goal is to find a public transport supply consisting of a line plan, a timetable, a routing of the passengers, and a vehicle schedule that is feasible and minimizes a weighted sum of the operational costs and the passengers' travel times. Here, the operational costs contain duration-based and distance-based costs for trips covering lines and relocations between them as well as costs depending on the number of vehicles needed. The structure of the integrated IP model is the following:

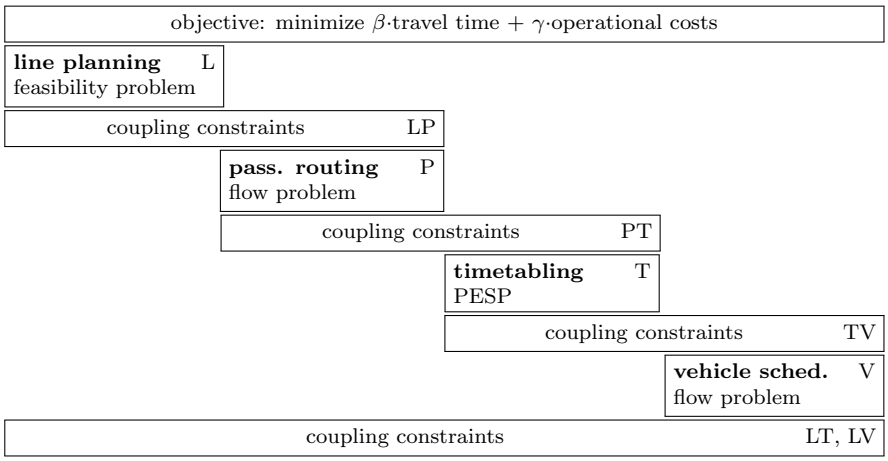


Figure 1: Structure of the integrated line planning, timetabling, and vehicle scheduling problem with passenger routing.

We abbreviate the four planning stages by L, P, T and V, and hence classify the coupling constraints as LP, LT, LV, PT and TV. These classes are again used in the IP model to simplify the presentation of the constraints.

We now develop the integrated integer program in more detail. In order to keep the complexity of the model manageable, we introduce the following assumptions:

- A public transportation network (PTN) (V, E) consisting of stops V and direct connections (e.g., tracks) E between the nodes is given.
- Lines are chosen from a fixed line pool $\mathcal{L}^0$ and are operated with a frequency of one if used. This is encoded in the Boolean variable f_l, $l \in \mathcal{L}^0$.
- Lower and upper frequency bounds $f_e^{\min}, f_e^{\max}$ on the edges $e \in E$ are provided. The lower frequency bounds $f_e^{\min}$ guarantee a prespecified service quality while the upper frequency bounds $f_e^{\max}$ deal with operational issues such as headway constraints.
- The timetable is periodic with period length T.
- origin-destination (OD) pairs are routed as a unit on shortest paths according to the travel time.
- The number of periods considered for vehicle scheduling is $p_{\max}$, we write the set of considered periods as $\mathcal{P} = \{1, \dots, p_{\max}\}$.
- The number of vehicles is not limited.
- There is one fixed depot where all vehicles start and end their journey.
- The minimal turnover time between the last event of line l_1 and the first event of line l_2 is given as L_{l_1,l_2}.

The model is based on an event-activity network (EAN) $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$, see (Serafini and Ukovich, 1989; Odijk, 1996), which is derived from the PTN and contains events and activities for all lines in the pool. The events represent departures and arrivals of vehicles at stops while the activities represent driving and waiting of vehicles as well as transferring of passengers between different lines.

$$
\begin{aligned}
\mathcal{E}^0 &= \mathcal{E}^0_{\text{arr}} \cup \mathcal{E}^0_{\text{dep}} \text{ where} \\
\mathcal{E}^0_{\text{arr}} &= \{(v, l, \text{arr}) \colon v \in l \cap V, l \in \mathcal{L}^0\} \\
\mathcal{E}^0_{\text{dep}} &= \{(v, l, \text{dep}) \colon v \in l \cap V, l \in \mathcal{L}^0\} \\
\mathcal{A}^0 &= \mathcal{A}^0_{\text{drive}} \cup \mathcal{A}^0_{\text{wait}} \cup \mathcal{A}^0_{\text{trans}} \text{ where} \\
\mathcal{A}^0_{\text{drive}} &= \{((v_1, l, \text{dep}), (v_2, l, \text{arr})) \colon \{v_1, v_2\} \in l \cap E, l \in \mathcal{L}^0\} \\
\mathcal{A}^0_{\text{wait}} &= \{((v, l, \text{arr}), (v, l, \text{dep})) \colon v \in l \cap V, l \in \mathcal{L}^0\} \\
\mathcal{A}^0_{\text{trans}} &= \{((v, l_1, \text{arr}), (v, l_2, \text{dep})) \colon v \in l_1 \cap l_2 \cap V, l_1, l_2 \in \mathcal{L}^0\}
\end{aligned}
$$

These events have to be scheduled according to the lower and upper bounds L_a, U_a, respectively, on the duration of the activities $a \in \mathcal{A}^0$. Therefore, the variables π_i for the periodic time of the events $i \in \mathcal{E}^0$ and z_a for the modulo parameters on the activities $a \in \mathcal{A}^0$ are introduced. The auxiliary variables y_a are used to decide whether all lines corresponding to activity $a \in \mathcal{A}^0$ are operated. Note that $\mathcal{A}^0(l_1, l_2)$ is the set of activities $a = (i, j)$ such that event i belongs to line l_1 and event j belongs to line l_2 while $\mathcal{A}^0(l)$ is the set of activities $a = (i, j)$ such that i or j belongs to l .

To correctly model the passenger routing, the network has to be extended further to include source and target nodes for all OD pairs which correspond to nodes in the underlying infrastructure network as well as activities connecting these special events to the rest of the EAN. These new events need not be scheduled in the timetable. Along the lines of (Schmidt, 2014), we get an extended event-activity network $\bar{\mathcal{N}} = (\bar{\mathcal{E}}, \bar{\mathcal{A}})$ with

$$
\begin{aligned}
\bar{\mathcal{E}} &= \mathcal{E}^0 \cup \mathcal{E}^0_{\text{OD}} \\
\mathcal{E}^0_{\text{OD}} &= \{(u, v, \text{source}), (u, v, \text{target}) \colon u, v \in V\} \\
\bar{\mathcal{A}} &= \mathcal{A}^0 \cup \mathcal{A}^0_{\text{to}} \cup \mathcal{A}^0_{\text{from}} \\
\mathcal{A}^0_{\text{to}} &= \{((u, v, \text{source}), (u, l, \text{dep})) \colon u \in l \cap V, u, v \in V\} \\
\mathcal{A}^0_{\text{from}} &= \{((v, l, \text{arr}), (u, v, \text{target})) \colon v \in l \cap V, u, v \in V\}.
\end{aligned}
$$

Analogously to the definition of $\mathcal{A}^0(l)$, we define $\bar{\mathcal{A}}(l)$ as the set of activities $a = (i, j) \in \bar{\mathcal{A}}$ such that event i or event j belongs to line l.

As passengers are routed in the extended EAN $\bar{\mathcal{N}}$ in a flow model, we introduce a variable $p_a^{u,v}$ for each combination of OD pair (u, v) with $u, v \in V$ and activity $a \in \bar{\mathcal{A}}$ indicating whether the OD pair uses the activity.

For vehicle scheduling we introduce Boolean variables $x_{(p_1,l_1),(p_2,l_2)}$ to indicate whether the p_2-th driving of line l_2 is done by the vehicle that directly before that did the p_1-th driving of line l_1. Similarly, Boolean variables $x_{\text{depot},(p,l)}$ indicate whether the p-th driving of line l is done by a new vehicle from the depot and $x_{(p,l),\text{depot}}$ indicates whether the vehicle that did the p-th driving of line l is

going to the depot. To correctly describe the p-th driving of line l the following variables are used: d_l is the time it takes in the timetable to get from the first event in the line ($\texttt{first}(l)$) to the last event in the line ($\texttt{last}(l)$), $s_{p,l}, e_{p,l}$ is the start or end time of the p-th driving of line l, respectively.

The IP model can now be formulated in the following way, specifying the general structure given in Figure 1.

$$\begin{aligned}
\min \quad & \beta \cdot \sum_{u,v \in V} C_{u,v} \cdot \sum_{a=(i,j)\in\mathcal{A}^0} p_a^{u,v} \cdot (\pi_j - \pi_i + z_a \cdot T) && (1)\\
& + \gamma_1 \cdot \sum_{l\in\mathcal{L}^0} f_l \cdot d_l && (2)\\
& + \gamma_2 \cdot \sum_{l\in\mathcal{L}^0} f_l \cdot \text{length}_l && (3)\\
& + \gamma_3 \cdot \sum_{p_1\in\mathcal{P}} \sum_{l_1\in\mathcal{L}^0} \Big(\sum_{p_2\in\mathcal{P}} \sum_{l_2\in\mathcal{L}^0} \big(s_{(p_2,l_2)} - e_{(p_1,l_1)}\big) \cdot x_{(p_1,l_2),(p_2,l_2)} \\
& \qquad + x_{\text{depot},(p_1,l_1)} \cdot L_{\text{depot},l_1} + x_{(p_1,l_1),\text{depot}} \cdot L_{l_1,\text{depot}} \Big) && (4)\\
& + \gamma_4 \cdot \sum_{p_1\in\mathcal{P}} \sum_{l_1\in\mathcal{L}^0} \Big(\sum_{p_2\in\mathcal{P}} \sum_{l_2\in\mathcal{L}^0} \big(x_{(p_1,l_1),(p_2,l_2)} \cdot D_{l_1,l_2}\big) \\
& \qquad + x_{\text{dep},(p_1,l_1)} \cdot D_{\text{dep},l_1} + x_{(p_1,l_1),\text{dep}} \cdot D_{l_1,\text{dep}} \Big) && (5)\\
& + \gamma_5 \cdot \sum_{p\in\mathcal{P}} \sum_{l\in\mathcal{L}^0} x_{\text{depot},(p,l)} && (6)
\end{aligned}$$

$$\sum_{\substack{l\in\mathcal{L}^0:\\ e\in l}} f_l \geq f_e^{\min} \qquad e\in E \qquad \text{(L1)}$$

$$\sum_{\substack{l\in\mathcal{L}^0:\\ e\in l}} f_l \leq f_e^{\max} \qquad e\in E \qquad \text{(L2)}$$

$$\pi_j - \pi_i + z_a \cdot T \geq y_a \cdot L_a \qquad a=(i,j)\in\mathcal{A}^0 \qquad \text{(T1)}$$

$$\pi_j - \pi_i + z_a \cdot T \leq U_a + M\cdot(1-y_a) \qquad a=(i,j)\in\mathcal{A}^0 \qquad \text{(T2)}$$

$$y_a = f_{l_1}\cdot f_{l_2} \qquad a\in\mathcal{A}^0(l_1,l_2) \qquad \text{(LT1)}$$

$$A^{u,v}\cdot(p_a^{u,v})_{a\in\bar{\mathcal{A}}} = b^{u,v} \qquad u,v\in V \qquad \text{(P1)}$$

$$f_l \geq p_a^{u,v} \qquad u,v\in V, a\in\bar{\mathcal{A}}(l) \qquad \text{(LP1)}$$

$$d_l = \sum_{a=(i,j)\in\mathcal{A}^0(l,l)} (\pi_j-\pi_i+z_aT) l\in\mathcal{L}^0 \qquad \text{(TV1)}$$

$$s_{p,l} = p\cdot T + \pi_{\texttt{first}(l)} \qquad p\in\mathcal{P}, l\in\mathcal{L}^0 \qquad \text{(TV2)}$$

$$e_{p,l} = p\cdot T + \pi_{\texttt{first}(l)} + d_l \qquad p\in\mathcal{P}, l\in\mathcal{L}^0 \qquad \text{(TV3)}$$

$$\begin{aligned} s_{p_2,l_2} - e_{p_1,l_1} \geq\ & x_{(p_1,l_1),(p_2,l_2)}\cdot L_{l_1,l_2} \\ & - M'\cdot(1-x_{(p_1,l_1),(p_2,l_2)}) \end{aligned} \qquad p_1,p_2\in\mathcal{P}, l_1,l_2\in\mathcal{L}^0 \qquad \text{(V1)}$$

$$\begin{aligned} f_{l_2} =\ & \sum_{p_1\in\mathcal{P}}\sum_{l_1\in\mathcal{L}^0} x_{(p_1,l_1),(p_2,l_2)} \\ & + x_{\text{depot},(p_2,l_2)} \end{aligned} \qquad p_2\in\mathcal{P}, l_2\in\mathcal{L}^0 \qquad \text{(LV1)}$$

$$\begin{aligned} f_{l_1} =\ & \sum_{p_2\in\mathcal{P}}\sum_{l_2\in\mathcal{L}^0} x_{(p_1,l_1),(p_2,l_2)} \\ & + x_{(p_1,l_1),\text{depot}} \end{aligned} \qquad p_1\in\mathcal{P}, l_1\in\mathcal{L}^0 \qquad \text{(LV2)}$$

$$x_{(p,l),\bullet} \leq f_l \qquad p\in\mathcal{P}, l\in\mathcal{L}^0 \qquad \text{(LV3)}$$

$$x_{\bullet,(p,l)} \leq f_l \qquad p\in\mathcal{P}, l\in\mathcal{L}^0 \qquad \text{(LV4)}$$

$$\pi_i\in\{0,\dots,T-1\}, z_a\in\mathbb{Z}, y_a, f_l, p_a^{u,v}\in\{0,1\}, d_l, s_{p,l}, e_{p,l}\in\mathbb{N},$$
$$x_{(p_1,l_1),(p_2,l_2)}, x_{\text{depot},(p,l)}, x_{(p,l),\text{depot}}\in\{0,1\}$$

The objective function, see (1) to (6), minimizes a weighted sum of the travel time of the passengers, see (1), and the operational costs, see (2) to (6). While the travel time in (1) is weighted with β, the duration of trips in (2) and their length in (3) are weighted by γ_1 and γ_2, respectively. The duration of relocations between trips in (4) and their length in (5) are weighted by γ_3 and γ_4, respectively while the number of vehicles in (6) is weighted by γ_5. Linearizing the quadratic terms in (1) and (4) leads to adding further variables and constraints.

Equations (L1) and (L2) are the standard feasibility constraints for line planning, guaranteeing a prespecified service quality as well as a rough upper bound on the costs. Equations (T1), (T2), and (LT1) guarantee the feasibility of the timetable for the lines which are operated. The passenger flow is modeled by (P1) where a node-arc-incidence matrix is used and equation (LP1) ensures

that only arcs belonging to operated lines are used by passengers. Equations (TV1), (TV2), and (TV3) model the correct (aperiodic) time for the start and end of the trips which is used in equation (V1) to ensure that the time between two trips which are operated directly after one another is sufficiently large. The flow of the vehicles is modeled in equations (LV1) and (LV2) while equations (LV3) and (LV4) ensure that only trips belonging to operated lines are used. Note that constraint (LT1) can easily be linearized as product of two Boolean variables.

Note that the formulation given above can furthermore be extended to include the time slice model introduced in (Gattermann et al, 2016) in order to distribute the favored departure times of the passengers. This is omitted here to not further complicate the model.

3 Dantzig-Wolfe decompositions

In this section we apply a generic column generation approach to the integrated line planning, timetabling, and vehicle scheduling problem.

The structure presented in Section 2.2 can be exploited using the so-called *Dantzig-Wolfe decomposition* (DWD) (Dantzig and Wolfe, 1960): The problem is reformulated according to the given structure where each block is represented as a subproblem. Furthermore, a master problem has the task to select feasible solutions from each subproblem such that the coupling constraints are satisfied. Due to the exponentially large number of variables, this master problem is solved by *column generation*: variables are generated dynamically when solving the linear relaxation. Embedding this in a branch-and-bound algorithm yields *branch-and-price*. For an overview on column generation and branch-and-price, see e.g., (Desaulniers et al, 2005; Vanderbeck and Wolsey, 2010).

The above problem structure, consisting of the subproblems line planning, passenger routing, timetabling, and vehicle scheduling, seems to be the "canonical" one for applying a DWD. However, any structure that subdivides the coefficient matrix into blocks and coupling constraints is theoretically suitable for DWD. Here, two different *blocks* are independent from one another as they neither share variables nor constraints. (If *linking variables* are present, i.e., variables that are shared by two or more blocks, one can reformulate the problem by adding for each such variable a copy for each block it appears in, and then introducing coupling constraints that state that the variable copies must attain the same values, a so-called Langrangian decomposition.) Thus, a broad variety of structures exist that might be used to decompose the problem. The questions that arise are:

- What other decomposition structures do exist, apart from the canonical one?
- Is the canonical structure suited best for applying a DWD and performing branch-and-price? Or, if there exist other structures, does this decomposition-based solution approach perform better on them?
- Are there any properties that can serve as indicators of a good performance?

To find other decompositions than the canonical one, we use several structure detection algorithms, some of them described in (Bergner et al, 2015). Formally, a structure detection algorithm tries to find a mapping $C \to \mathbb{N}_0$, where C is the set of constraints. A constraint that is mapped to 0 is a coupling constraint, i.e., a constraint that belongs to no block, and thus is part of the master problem. Depending on the detection algorithm, the mapping either already guarantees that constraints mapped to the same integer form a block or blocks have to be formed by moving variables to the set of linking variables.

A key feature of the detection that we use is that the algorithms are allowed to determine *partial* structures $C \to \mathbb{N}_0 \cup \{\mathsf{open}\}$; i.e., a constraint can be left undecided (mapping it to open), and a partial structure that contains undecided constraints can then be completed by another algorithm. In the end, each complete structure (i.e. a structure with no undecided constraints left) usually has been detected by a combination of several of these detection algorithms, see (Gleixner et al, 2018) for more details. This increases the number of found structures and the chance to find suitable decompositions, but leaves the challenge to choose a "meaningful" one with which branch-and-price is expected to perform best.

Very roughly, structure detection proceeds in the following steps:

1. *Constraint classifiers* determine partitions of C, e.g., according to the number of variables and their coefficients. With these partitions, potential candidates for the number of blocks are determined.

2. Then, partial decompositions are built that only assign certain constraints to be coupling constraints, but leave the remainder open. This is done in the following ways:

   - by the above mentioned constraint classifiers;
   - by analyzing the *densities* of the constraints: Constraints with a high number of variables are assigned as coupling constraints.
   - by *graph partitioning*: The coefficient matrix $A \in \mathbb{R}^{m \times n}$ is modeled as a hypergraph in two different ways:
     - *hyper row graph*: Each node represents a column j, and a hyperedge $\{j : a_{ij} \neq 0\}$ for each row i is introduced;
     - *hyper row-column graph*: Each node represents a matrix entry (i, j) with $a_{ij} \neq 0$, and each row i is represented by a hyperedge $\{(i, j) : a_{ij} \neq 0\}$ containing its nonzero entries; analogously, there is a hyperedge for each column j containing its nonzero entries.

   Then, graph partitioning algorithms are applied on these graphs. These graph partitioners yield complete decompositions as well as partial decompositions which again only assign coupling constraints.

3. The partial decompositions are completed by looking for *connected components* on the remaining constraints.

4. Last, a *postprocessing* routine checks if coupling constraints can be assigned to blocks: If a coupling constraint only contains variables of one block, it will be moved to this block.

A overview on the detection algorithms we use is given in Table 1.

Step	Letter	Algorithm
1/2	c	constraint classification
2	a	graph partitioning on the hyper row-column graph
2	r	graph partitioning on the hyper row graph
3	C	searching connected components
3	d	detection by constraint densities
4	p	postprocessing

Table 1: Overview on detection algorithms. The algorithms c, a, and r derive a partial decomposition according to Step 2. They can be followed by the algorithms C and d, see Step 3. The postprocessing algorithm p can be performed after each of the other algorithms.

4 Computational experiments

The structure detection is implemented in the generic branch-and-price solver GCG (Gamrath and Lübbecke, 2010) which we use in a development version based on version 2.1.4. GCG is an extension to SCIP, used in version 4.0, see (Gleixner et al, 2018), a solver for mixed integer programs that also serves as a framework for branch-price-and-cut.

We applied the above structure detection scheme on a small example instance depicted in Figure 2. Since SCIP comes with various presolving routines which may change the problem formulation and in particular add new constraints, the detection scheme was applied twice: first on the original IP formulation, then, after presolving, again on potentially newly added constraints. In total, this yielded 75 decompositions.

We evaluated each decomposition w.r.t. its computational performance within branch-and-price: Therefore, we tried to solve the root LP relaxation within a time limit of one hour. Note that this is the LP relaxation of the master problem, i.e., the subproblems are solved with integrality constraints but the combination of columns found by the subproblems to a solution of the master problem can be rational instead of integral. The computations were performed on an Intel(R) Core(TM) i7-2600 CPU at 3.6 GHz, with 16 GB RAM and 8 MB cache, running on openSUSE Leap 42.2 with Linux kernel 4.4. The results are shown in Table 2; for each decomposition, it shows the involved algorithms; moreover, the relative block and border area, the time and number of LP iterations needed to solve the root LP relaxation, and the gap between the dual bound and the optimal solution value of the integrated IP. Here, the optimal value of the IP can be used to compute the gap as the integrated problem is small enough such that it can be solved to optimality by commercial solvers.

4.1 Canonical decomposition

At first, we consider the "canonical" decomposition structure which uses the subproblems line planning, timetabling, vehicle scheduling, and passenger routing from the sequential process as blocks. It is depicted in Figure 3.

Figure 3a is a reordering of the schematic representation of the matrix struc-

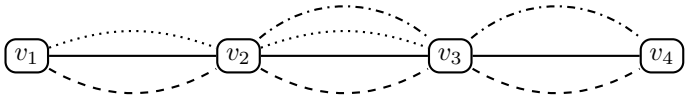
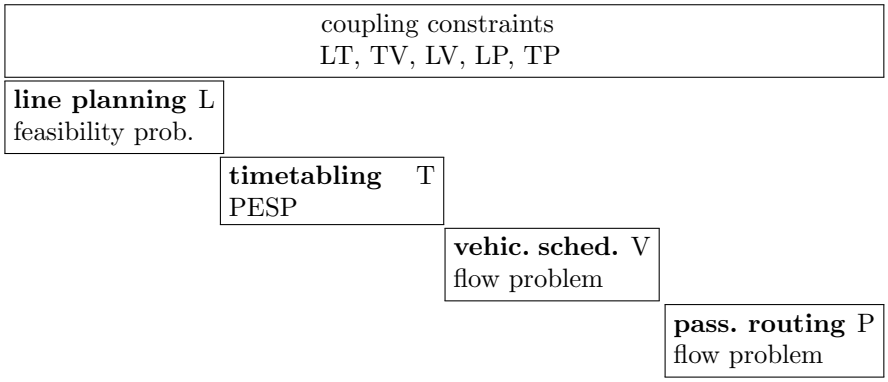
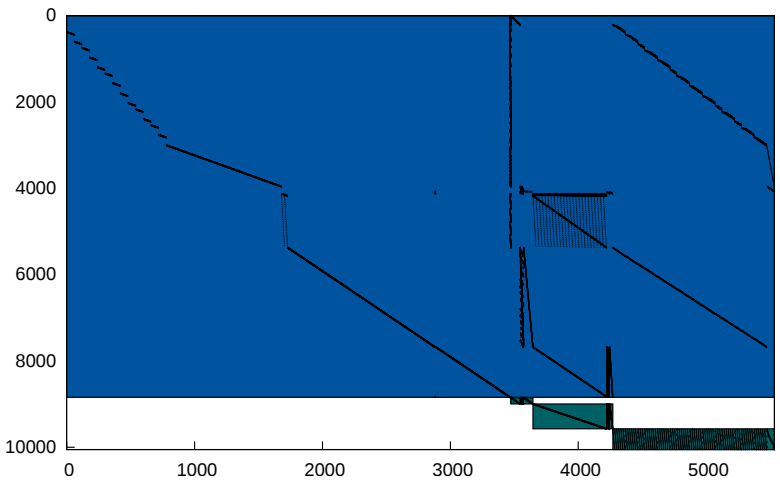


Figure 2: Data set used for testing. The solid lines represent the infrastructure network while the dashed lines represent the lines of the line pool.



(a) Schematic representation of the matrix structure, reordering of Figure 1.



(b) Actual matrix structure. The variables are numbered on the x-axis and the constraints on the y-axis. The small blocks (dark green area) represent the subproblems timetabling, vehicle scheduling, and passenger routing, with an additional block for line planning which consists of 12 constraints and cannot be seen. The large dark blue area represents the large number of coupling constraints.

Figure 3: Canonical decomposition of the integrated line planning, timetabling, and vehicle scheduling problem with passenger routing.

ture given in Figure 1 while Figure 3b represents the actual matrix structure

for the instance given in Figure 2. The large area at the top represents the coupling constraints which clearly make up most of the coefficient matrix thus making it hard to find subproblems which can be solved independently. Also note the large number of variables which do not occur in any of the blocks. These are auxiliary variables used for linearizations of constraints (LT1) and of the objective and are not explicitly mentioned in the model.

When solving the problem in this canonical form by SCIP, the LP at the root node of the branch-and-price tree cannot even be solved within the time limit. Due to the then poor lower bound, the gap is still at 5182.32 % which is far from optimal.

4.2 Influence of detection algorithms

Therefore, we now consider other decompositions found by GCG. In the following, we denote each combination of detection algorithms by the letters in Table 1. E.g., apCp means applying graph partitioning and postprocessing on the un-presolved problem, then searching connected components and again performing postprocessing on newly added constraints after presolving.

Figure 4 shows the solvability of the matrix structures found by the (combinations of) different algorithms indicated by the gap after solving the root node of the branch-and-price tree.

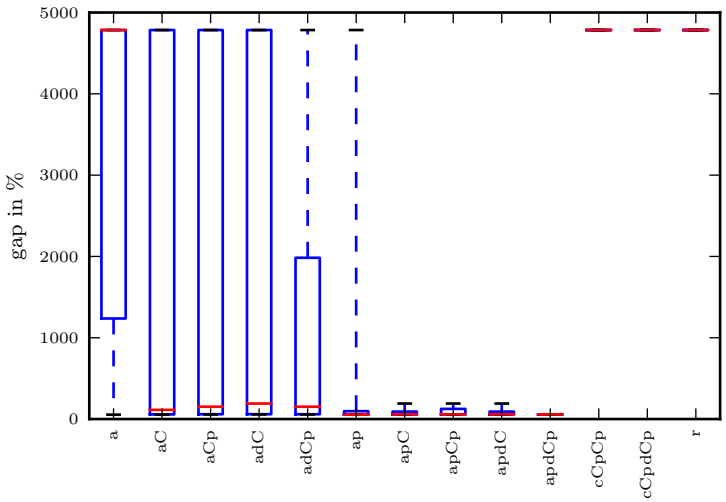


Figure 4: Box and whiskers plot of the performance of the different decomposition algorithms. The algorithms listed on the x-axis are combinations of the detection algorithms given in Table 1. The boxes mark the 25th to 75th percentile while the whiskers mark the minimal and maximal values. The median is depicted by a red line. Here, the performance is measured as the gap after solving the root node of the branch-and-price tree which is given on the y-axis.

Figure 4 suggests that graph partitioning algorithms on hyper row-column graphs combined with connected components are better suited for the integrated

line planning, timetabling, and vehicle scheduling problem then algorithms using constraint classification or graph partitioning on hyper row graphs.

Especially the algorithms apC, apCp, apdC and apdCp lead to good structures. A typical example of a decomposition found by these algorithms is depicted in Figure 5. Such decompositions are called *arrowhead matrices* due to their shape. Intuitively, these decompositions seem to be easier to solve due to the low number of coupling constraints and variables combined with independent blocks of reasonable sizes.

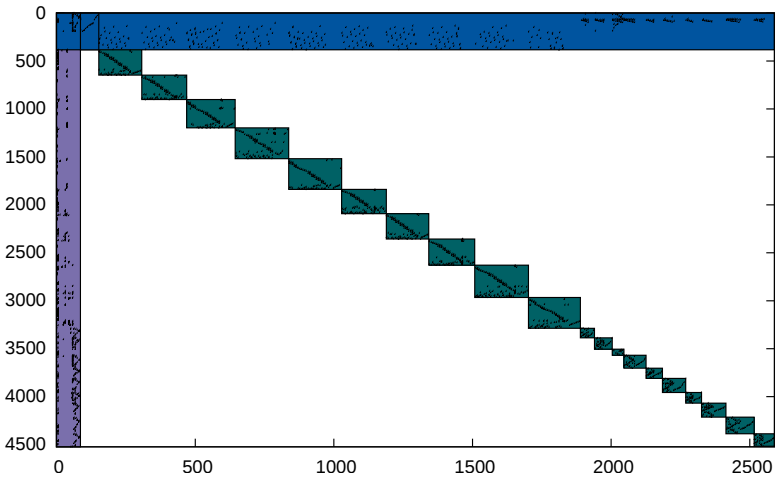
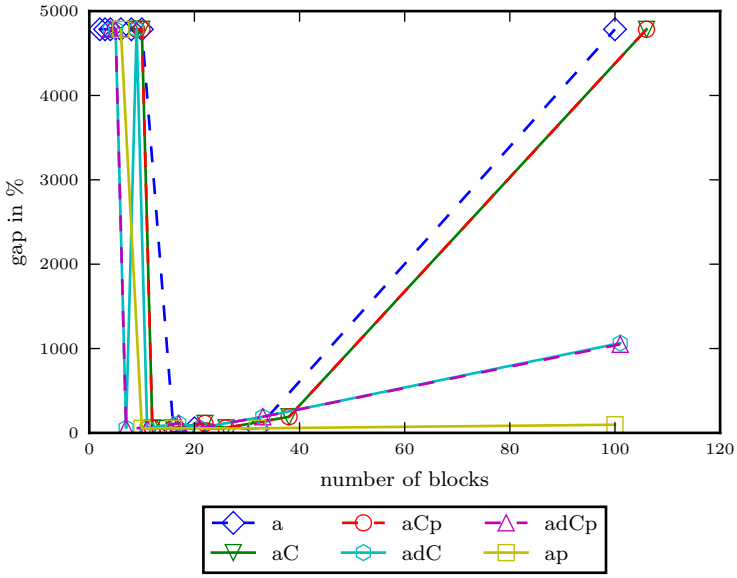


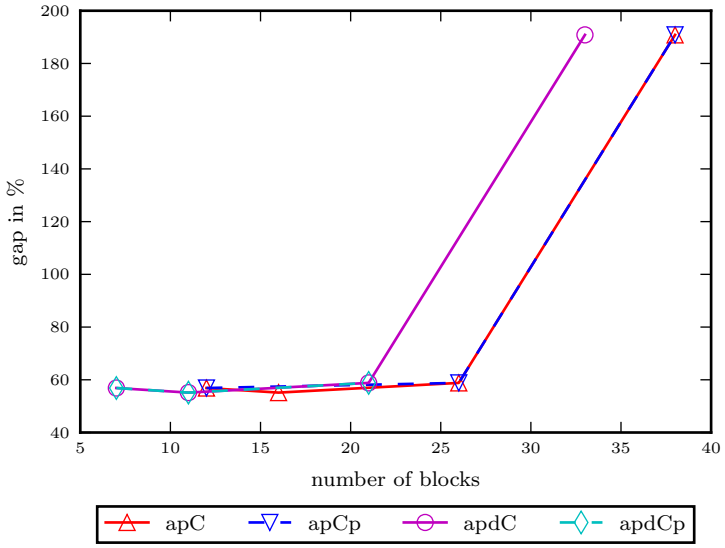
Figure 5: Example for an arrowhead matrix found by the algorithm apC with 26 blocks. The variables are numbered on the x-axis and the constraints on the y-axis. The dark blue area represents the coupling constraints, the purple area represents the linking variables, and the dark green area represents the blocks.

4.3 Influence of the number of blocks

Figure 6 shows the influence of the number of blocks on the solvability. While good decompositions could be found for a large span of the number of blocks, high and low numbers of blocks can also lead to bad decompositions while a medium number of around 20 to 30 blocks is more promising.



(a) Algorithms a, aC, aCp, adC, adCp and ap, see Table 1.



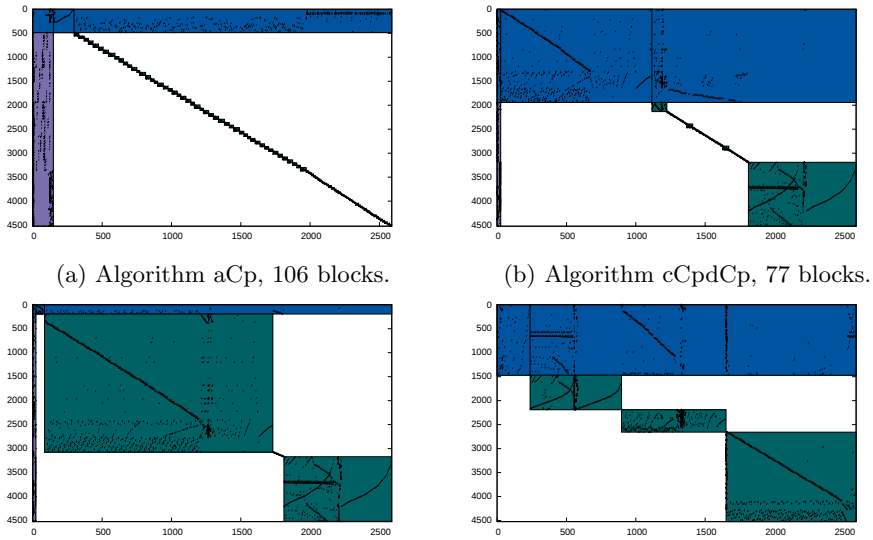
(b) Algorithms apC, apCp, apdC, apdCp, see Table 1.

Figure 6: Influence of the number of blocks on the performance of column generation. Here, the performance is measured as the gap after solving the root node of the branch-and-price tree which is given on the y-axis.

Note that the scale of Figure 6a and 6b varies as Figure 6b only contains "good" algorithms which lead to a gap of less than 200 percent. Figure 6a also

shows an effect which occurred for all decompositions considered here: The gap is either acceptably small (up to 200%) or the root node LP could not be solved, leading to a gap of several thousand percent.

Examples for decompositions with a large gap are given in Figure 7. They either feature many very small blocks (Figure 7a and 7b) or few large blocks (Figure 7d) or a combination of both (Figure7c). Medium-sized blocks seem to be more promising.



(a) Algorithm aCp, 106 blocks.

(b) Algorithm cCpdCp, 77 blocks.

(c) Algorithm cCpCp, 24 blocks.

(d) Algorithm r, 3 blocks.

Figure 7: Decompositions with a large gap. The variables are numbered on the x-axis and the constraints on the y-axis. The dark blue areas represent the coupling constraints, the purple areas represent the linking variables and the dark green areas represent the blocks.

4.4 Block and border scores

To further characterize the decompositions we consider the *block score*

$$\text{block} = \frac{\sum_{k=1}^{K} m_k \cdot n_k}{m \cdot n} \tag{7}$$

and the *border score*

$$\text{border} = \frac{m_0 \cdot n + (\sum_{k=1}^{K} m_k) \cdot n_0}{m \cdot n}, \tag{8}$$

where K is the number of blocks, m and n are the total number of constraints and variables, respectively, m_k and n_k are the number of constraints and variables in block k, respectively, m_0 is the number of coupling constraints and n_0 the number of linking variables.

In an IP model structure, these two scores indicate the relative block and border area, respectively. Our expectation is that decompositions with smaller

scores lead to a better computational performance, see (Bergner et al, 2015; Gleixner et al, 2018).

Figure 8 shows the block and border scores for each decomposition. The canonical decomposition which is depicted by a star differentiates itself by a very high border score compared to all other decompositions considered here and one of the lowest block scores. This could already be seen in Figure 3 in the large number of coupling constraints. Decompositions for which the root LP can be solved such that the gap is less than 200 % all feature both low block and border scores. Nevertheless, low block and border scores are only an indicator and no guarantee for good solvability, see Bergner et al (2015) for a general discussion of such measures.

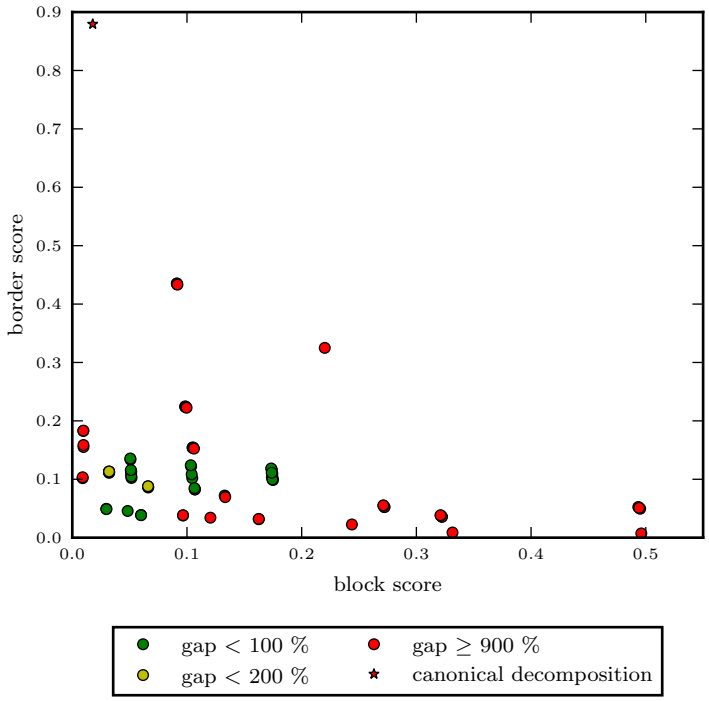


Figure 8: Comparing different decompositions regarding the block score and the border score, see equations (7) and (8), respectively.

5 Outlook

In this paper, we developed an integrated formulation for line planning, timetabling, and vehicle scheduling with passenger routing. We analyzed the structure of the resulting integer program with respect to possible decompositions.

The analysis of different matrix structure and their influence on the solvability of the integrated line planning, timetabling, and vehicle scheduling problem presented here can only be a first step to understanding this problem. Although we show that the canonical decomposition is far from being an optimal one to

solve the integrated problem it is hard to determine which decompositions are suited better to this end. The best indicator so far is a small block score and a small border score at the same time. Therefore, extensive computational studies are needed to confirm this correlation and find further indicators for good decompositions.

To improve the efficiency of solving different decompositions, the combination with existing heuristics for problems in public transport planning would be interesting. Whenever a subproblem is detected that can be interpreted as one of the problems stemming from the original planning process (i.e., line planning, timetabling, or vehicle scheduling), well researched specialized heuristics can be used to solve this subproblem. From this we expect a speed-up of the computations, leading to better results for solving the integrated problem.

In addition to better understanding the decompositions, it would be interesting to further extend the scope of the integrated problem, e.g., by adding robustness aspects. It is of particular interest which robustness concept makes sense in this context (for timetabling, see (Goerigk and Schöbel, 2010) for a study of different robustness concepts). In our setting, robustness can be considered from a passengers' and an operators' point of view. While robust passenger paths depend mainly on transfers which should not break due to small delays, robust vehicle schedules need should allow for a certain number of delayed vehicles before new ones have to be scheduled.

Furthermore, the integrated model could be extended by additional planning stages. On the one hand, adding network design aspects such as stop location to the formulation extends the scope of the model in regard to planning completely new public transport supplies. On the other hand, adding crew scheduling leads to an even better approximation of the operational costs. This, however, increases the computational challenge of the model even more and has to be combined with solution approaches for reducing the problem size.

Acknowledgements

We would like to thank Michael Bastubbe from RWTH Aachen University for designing and implementing the structure detection scheme in GCG, as well as contributing a number of detection algorithms.

Appendix

Det.	*K*	*block*	*border*	**# LP iters**	**Time (s)**	**Gap (%)**
a	2	0.50	0.01	274	3600.00	4784.50
a	4	0.24	0.02	1913	3600.00	4784.50
a	8	0.12	0.03	16564	3600.00	4784.50
a	16	0.06	0.04	29172	223.20	54.30
a	32	0.03	0.05	48081	97.90	54.30
a	3	0.33	0.01	268	3600.00	4784.50
a	10	0.10	0.04	40097	3600.00	4784.50

Continue next page

Det.	K	*block*	*border*	**# LP iters**	**Time (s)**	**Gap (%)**
a	6	0.16	0.03	18243	3600.00	4784.50
a	20	0.05	0.05	46721	160.80	54.30
a	100	0.01	0.10	1420136	481.10	4784.50
aC	10	0.27	0.06	276	3600.00	4784.50
aC	14	0.13	0.07	147856	2853.20	54.40
aC	22	0.07	0.09	187441	636.70	113.20
aC	38	0.03	0.11	251646	739.30	190.80
aC	9	0.32	0.04	273	3600.00	4784.50
aC	12	0.17	0.12	201955	1415.90	57.20
aC	16	0.10	0.12	544836	904.40	56.80
aC	26	0.05	0.14	2553719	2829.10	59.80
aC	106	0.01	0.18	1589003	1463.90	4784.50
aCp	10	0.27	0.05	276	3600.00	4784.50
aCp	22	0.07	0.09	224273	635.90	113.20
aCp	38	0.03	0.11	265512	727.40	190.80
aCp	9	0.32	0.04	273	3600.00	4784.50
aCp	12	0.17	0.10	290977	1639.60	57.20
aCp	16	0.10	0.11	577623	1019.20	56.60
aCp	26	0.05	0.12	968633	1266.70	59.70
aCp	106	0.01	0.16	3211795	3423.60	4784.50
adC	5	0.27	0.05	271	3600.00	4784.50
adC	9	0.13	0.07	108036	3600.00	4784.50
adC	17	0.07	0.09	138008	585.00	113.20
adC	33	0.03	0.11	206704	609.60	190.80
adC	4	0.32	0.04	268	3600.00	4784.50
adC	7	0.17	0.12	285443	1414.80	57.20
adC	11	0.10	0.12	574388	1030.90	56.80
adC	21	0.05	0.13	1876435	2067.10	59.80
adC	101	0.01	0.18	1232117	1703.80	1064.30
adCp	5	0.27	0.05	271	3600.00	4784.50
adCp	17	0.07	0.09	154553	543.40	113.20
adCp	33	0.03	0.11	259472	805.80	190.80
adCp	4	0.32	0.04	268	3600.00	4784.50
adCp	7	0.17	0.10	283497	1813.00	57.20
adCp	11	0.10	0.10	699098	1081.20	56.50
adCp	21	0.05	0.11	721362	1072.60	59.70
adCp	101	0.01	0.16	2822062	3518.00	1050.20
ap	16	0.06	0.04	33182	234.70	54.30
ap	32	0.03	0.05	44427	99.00	54.30
ap	10	0.10	0.04	38957	2621.40	54.30
ap	6	0.16	0.03	23019	3600.00	4784.50
ap	100	0.01	0.10	326158	385.80	96.70
apC	38	0.03	0.11	212859	620.80	190.80
apC	12	0.17	0.11	303335	1840.00	56.90
apC	16	0.11	0.08	228118	843.70	55.10
apC	26	0.05	0.12	1179709	1773.90	58.80

Continue next page

Det.	K	*block*	*border*	**# LP iters**	**Time (s)**	**Gap (%)**
apC	106	0.01	0.12	1642579	2619.60	910.40
apCp	38	0.03	0.11	247731	760.40	190.80
apCp	12	0.17	0.10	332521	1776.50	56.90
apCp	26	0.05	0.10	577782	1052.80	58.80
apCp	106	0.01	0.12	1800497	2891.40	910.40
apdC	33	0.03	0.11	235631	670.50	190.80
apdC	7	0.17	0.11	330598	1882.80	56.90
apdC	11	0.11	0.08	348440	836.60	55.10
apdC	21	0.05	0.11	1327333	1828.00	58.80
apdC	101	0.01	0.12	740547	1826.10	910.40
apdCp	7	0.17	0.10	279236	2162.70	56.90
apdCp	11	0.11	0.08	315679	908.80	55.10
apdCp	21	0.05	0.10	571431	978.20	58.80
apdCp	101	0.01	0.12	994508	1950.80	910.40
cCpCp	24	0.50	0.05	292	3600.00	4784.50
cCpCp	24	0.49	0.05	902	3600.00	4784.50
cCpCp	24	0.11	0.15	308	3600.00	4784.50
cCpCp	24	0.49	0.05	472	3600.00	4784.50
cCpCp	24	0.11	0.15	322	3600.00	4784.50
cCpdCp	19	0.10	0.15	392	3600.00	4784.50
cCpdCp	19	0.10	0.15	1460	3600.00	4784.50
r	3	0.22	0.33	10928	3600.00	4784.50

Table 2: Detailed results for each decomposition

References

Bergner M, Caprara A, Ceselli A, Furini F, Lübbecke ME, Malaguti E, Traversi E (2015) Automatic Dantzig–Wolfe reformulation of mixed integer programs. Mathematical Programming 149(1-2):391–424

Borndörfer R, Hoppmann H, Karbstein M (2016) Passenger routing for periodic timetable optimization. Public Transport DOI 10.1007/s12469-016-0132-0

Bunte S, Kliewer N (2009) An overview on vehicle scheduling models. Public Transport 1(4):299–317

Burggraeve S, Bull S, Vansteenwegen P, Lusby R (2017) Integrating robust timetabling in line plan optimization for railway systems. Transportation Research Part C: Emerging Technologies 77:134–160

Bussieck M, Winter T, Zimmermann U (1997) Discrete optimization in public rail transport. Mathematical Programming 79(1-3):415–444

Cadarso L, Marín Á (2012) Integration of timetable planning and rolling stock in rapid transit networks. Annals of Operations Research 199(1):113–135

Dantzig GB, Wolfe P (1960) Decomposition principle for linear programs. Operations Research 8(1):101–111

Desaulniers G, Hickman M (2007) Public transit. Handbooks in Operations Research and Management Science 14:69–127

Desaulniers G, Desrosiers J, Solomon MM (eds) (2005) Column Generation. Springer

Fonseca J, van der Hurk E, Roberti R, Larsen A (2018) A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling. Transportation Research Part B: Methodological 109:128–149

Gamrath G, Lübbecke ME (2010) Experiments with a generic Dantzig-Wolfe decomposition for integer programs. In: Experimental Algorithms, Springer, LNCS, vol 6049, pp 239–252

Gattermann P, Großmann P, Nachtigall K, Schöbel A (2016) Integrating passengers' routes in periodic timetabling: A SAT approach. In: ATMOS 2016, OpenAccess Series in Informatics (OASIcs), vol 54, pp 1–15, URL http://drops.dagstuhl.de/opus/volltexte/2016/6527

Gleixner A, Bastubbe M, Eifler L, Gally T, Gamrath G, Gottwald RL, Hendel G, Hojny C, Koch T, Lübbecke ME, Maher SJ, Miltenberger M, Müller B, Pfetsch ME, Puchert C, Rehfeldt D, Schlösser F, Schubert C, Serrano F, Shinano Y, Viernickel JM, Walter M, Wegscheider F, Witt JT, Witzig J (2018) The SCIP Optimization Suite 6.0. ZIB-Report 18-26, Zuse-Institute Berlin

Goerigk M, Schöbel A (2010) An empirical analysis of robustness concepts for timetabling. In: Erlebach T, Lübbecke M (eds) Proceedings of ATMOS10, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, OpenAccess Series in Informatics (OASIcs), vol 14, pp 100–113, DOI http://dx.doi.org/10.4230/OASIcs.ATMOS.2010.100, URL http://drops.dagstuhl.de/opus/volltexte/2010/2753

Guihaire V, Hao J (2008) Transit network design and scheduling: A global review. Transportation Research Part A: Policy and Practice 42(10):1251–1273

Guihaire V, Hao JK (2010) Transit network timetabling and vehicle assignment for regulating authorities. Computers & Industrial Engineering 59(1):16–23

van den Heuvel A, van den Akker J, van Kooten M (2008) Integrating timetabling and vehicle scheduling in public bus transportation. Tech. rep., Utrecht University

Huisman D, Kroon L, Lentink R, Vromans M (2005) Operations research in passenger railway transportation. Statistica Neerlandica 59(4):467–497

Ibarra-Rojas O, Rios-Solis Y (2011) Integrating synchronization bus timetabling and single-depot single-type vehicle scheduling. In: ORP3 Meeting, Cadiz

Kaspi M, Raviv T (2013) Service-oriented line planning and timetabling for passenger trains. Transportation Science 47(3):295–311

Kinder M (2008) Models for periodic timetabling. Master's thesis, Technische Universität Berlin

Liebchen C (2007) Periodic timetable optimization in public transport. Springer

Liebchen C (2008) Linien-, Fahrplan-, Umlauf- und Dienstplanoptimierung: Wie weit können diese bereits integriert werden? In: Heureka'08

Lusby R, Larsen J, Ehrgott M, Ryan D (2011) Railway track allocation: models and methods. OR Spectrum 33(4):843–883

Michaelis M, Schöbel A (2009) Integrating line planning, timetabling, and vehicle scheduling: A customer-oriented approach. Public Transport 1(3):211–232

Odijk M (1996) A constraint generation algorithm for the construction of periodic railway timetables. Transportation Research Part B: Methodological 30(6):455–464

Pätzold J, Schiewe A, Schiewe P, Schöbel A (2017) Look-Ahead Approaches for Integrated Planning in Public Transportation. In: D'Angelo G, Dollevoet T (eds) 17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, OpenAccess Series in Informatics (OASIcs), vol 59, pp 17:1–17:16

Petersen H, Larsen A, Madsen O, Petersen B, Ropke S (2013) The Simultaneous Vehicle Scheduling and Passenger Service Problem. Transportation Science 47(4):603–616

Rittner M, Nachtigall K (2009) Simultane Liniennetz- und Fahrlagenoptimierung. Der Eisenbahningenieur

Robenek T, Azadeh S, Maknoon Y, Bierlaire M (2017) Hybrid cyclicity: Combining the benefits of cyclic and non-cyclic timetables. Transportation Research Part C: Emerging Technologies 75:228–253

Schiewe A, Schiewe P (2018) An Iterative Approach for Integrated Planning in Public Transportation. Tech. rep., Georg-August-Universität Göttingen, working Paper

Schiewe P (2018) Integrated optimization in public transport planning. PhD thesis, Georg-August-Universität Göttingen

Schmid V, Ehmke JF (2015) Integrated timetabling and vehicle scheduling with balanced departure times. OR Spectrum 37(4):903–928

Schmidt D (2005) Linien- und Taktfahrplanung - Ein integrierter Optimierungsansatz. Master's thesis, Technische Universität Berlin, in German

Schmidt M (2014) Integrating Routing Decisions in Public Transportation Problems, Optimization and Its Applications, vol 89. Springer

Schmidt M, Schöbel A (2015a) The complexity of integrating routing decisions in public transportation models. Networks 65(3):228–243

Schmidt M, Schöbel A (2015b) Timetabling with passenger routing. OR Spectrum 37:75–97

Schöbel A (2012) Line planning in public transportation: models and methods. OR Spectrum 34(3):491–510

Schöbel A (2017) An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. Transportation Research C 74:348–365

Serafini P, Ukovich W (1989) A mathematical model for periodic scheduling problems. SIAM Journal on Discrete Mathematics 2(4):550–581

Siebert M, Goerigk M (2013) An experimental comparison of periodic timetabling models. Computers & Operations Research 40(10):2251–2259

Vanderbeck F, Wolsey LA (2010) Reformulation and decomposition of integer programs. In: 50 Years of Integer Programming 1958-2008, Springer, pp 431–502

Yue Y, Han J, Wang S, Liu X (2017) Integrated Train Timetabling and Rolling Stock Scheduling Model Based on Time-Dependent Demand for Urban Rail Transit. Computer-Aided Civil and Infrastructure Engineering 32(10):856–873