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; Robeneck et al, 2017) while line planning and timetabling are considered integratedly in (Schmidt, 2005; Liebchen, 2008; Rüttner 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; Robeneck 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:
Objective: minimize $\beta$-travel time + $\gamma$-operational costs

<table>
<thead>
<tr>
<th>Planning Stage</th>
<th>Description</th>
<th>Coupling Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Planning</td>
<td>Feasibility problem</td>
<td>LP</td>
</tr>
<tr>
<td>Passenger Routing</td>
<td>Flow problem</td>
<td>PT</td>
</tr>
<tr>
<td>Timetabling</td>
<td>PESP</td>
<td>TV</td>
</tr>
<tr>
<td>Vehicle Sched.</td>
<td>Flow problem</td>
<td>LT, LV</td>
</tr>
</tbody>
</table>

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 $L^0$ and are operated with a frequency of one if used. This is encoded in the Boolean variable $f_l$, $l \in L^0$.
- Lower and upper frequency bounds $f^{\text{min}}_e, f^{\text{max}}_e$ on the edges $e \in E$ are provided. The lower frequency bounds $f^{\text{min}}_e$ guarantee a prespecified service quality while the upper frequency bounds $f^{\text{max}}_e$ 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 $p_{\text{max}}$, we write the set of considered periods as $P = \{1, \ldots, p_{\text{max}}\}$.
- 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) \( N^0 = (E^0, A^0) \), 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.

\[
E^0 = E^0_{arr} \cup E^0_{dep} \text{ where } \\
E^0_{arr} = \{(v, l, \text{arr}): v \in l \cap V, l \in L^0\} \\
E^0_{dep} = \{(v, l, \text{dep}): v \in l \cap V, l \in L^0\} \\
A^0 = A^0_{\text{drive}} \cup A^0_{\text{wait}} \cup A^0_{\text{trans}} \text{ where } \\
A^0_{\text{drive}} = \{((v_1, l, \text{dep}), (v_2, l, \text{arr})): \{v_1, v_2\} \in l \cap E, l \in L^0\} \\
A^0_{\text{wait}} = \{((v, l, \text{arr}), (v, l, \text{dep})): v \in l \cap V, l \in L^0\} \\
A^0_{\text{trans}} = \{((v_1, l_1, \text{arr}), (v, l_2, \text{dep})): v \in l_1 \cap l_2 \cap V, l_1, l_2 \in L^0\}
\]

These events have to be scheduled according to the lower and upper bounds \( L_a, U_a \), respectively, on the duration of the activities \( a \in A^0 \). Therefore, the variables \( \pi_i \) for the periodic time of the events \( i \in E^0 \) and \( z_a \) for the modulo parameters on the activities \( a \in A^0 \) are introduced. The auxiliary variables \( y_a \) are used to decide whether all lines corresponding to activity \( a \in A^0 \) are operated. Note that \( A^0(l_1, l_2) \) 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 \( A^0(l) \) 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 \( \tilde{N} = (\tilde{E}, \tilde{A}) \) with

\[
\tilde{E} = E^0 \cup E_{OD}^0 \text{ where } \\
E_{OD}^0 = \{(u, v, \text{source}), (u, v, \text{target}): u, v \in V\} \\
\tilde{A} = A^0 \cup A_{to}^0 \cup A_{from}^0 \text{ where } \\
A_{to}^0 = \{((u, v, \text{source}), (u, l, \text{dep})): u \in l \cap V, u, v \in V\} \\
A_{from}^0 = \{((v, l, \text{arr}), (u, v, \text{target})): v \in l \cap V, u, v \in V\}.
\]

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

As passengers are routed in the extended EAN \( \tilde{N} \) in a flow model, we introduce a variable \( p^u_{vu} \) for each combination of OD pair \( (u, v) \) with \( u, v \in V \) and activity \( a \in \tilde{A} \) 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 ($\text{first}(l)$) to the last event in the line ($\text{last}(l)$), $s_{p,l}, e_{p,l}$ 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.

$$
\min \beta \cdot \sum_{u,v \in V} C_{u,v} \cdot \sum_{a=(i,j) \in A^u} p_a^{u,v} \cdot (\pi_j - \pi_i + z_a \cdot T)
+ \gamma_1 \cdot \sum_{l \in L^0} f_l \cdot d_l \tag{1}
+ \gamma_2 \cdot \sum_{l \in L^0} f_l \cdot \text{length}_l \tag{2}
+ \gamma_3 \cdot \sum_{p_1 \in P} \sum_{l_1 \in L^0} \left( \sum_{p_2 \in P} \sum_{l_2 \in L^0} \left( s_{p_2,l_2} - e_{p_1,l_1} \right) \cdot x_{(p_1,l_1),(p_2,l_2)} \right)
+ x_{\text{depot},(p_1,l_1)} \cdot L_{\text{depot},l_1} + x_{(p_1,l_1),\text{depot}} \cdot L_{l_1,\text{depot}} \tag{3}
+ \gamma_4 \cdot \sum_{p_1 \in P} \sum_{l_1 \in L^0} \left( \sum_{p_2 \in P} \sum_{l_2 \in L^0} \left( x_{(p_1,l_1),(p_2,l_2)} \cdot D_{l_1,l_2} \right) \right)
+ x_{\text{dep},(p_1,l_1)} \cdot D_{\text{dep},l_1} + x_{(p_1,l_1),\text{dep}} \cdot D_{l_1,\text{dep}} \tag{4}
+ \gamma_5 \cdot \sum_{p \in P} \sum_{l \in L^0} x_{\text{depot},(p,l)} \tag{5}
+ \gamma_6 \cdot \sum_{p \in P} \sum_{l \in L^0} x_{\text{dep},(p,l)} \tag{6}
$$
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 \( \beta \), the duration of trips in (2) and their length in (3) are weighted by \( \gamma_1 \) and \( \gamma_2 \), respectively. The duration of relocations between trips in (4) and their length in (5) are weighted by \( \gamma_3 \) and \( \gamma_4 \), respectively while the number of vehicles in (6) is weighted by \( \gamma_5 \). 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 Lagrangian 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 $C \rightarrow \mathbb{N}_0$, 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 $C \rightarrow \mathbb{N}_0 \cup \{\text{open}\}$; 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.
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.

<table>
<thead>
<tr>
<th>Coupling constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>LT, TV, LV, LP, TP</td>
</tr>
</tbody>
</table>

(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.](image)

### 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.
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 (Figure 7c). Medium-sized blocks seem to be more promising.

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

$$\text{block} = \frac{\sum_{k=1}^{K} m_k \cdot n_k}{m \cdot n}$$  \hspace{1cm} (7)

and the border score

$$\text{border} = m_0 \cdot n + \frac{\sum_{k=1}^{K} m_k \cdot n_k}{m \cdot n} \cdot n_0,$$  \hspace{1cm} (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.

\[
\begin{align*}
\text{block score} & = \text{equation (7)} \\
\text{border score} & = \text{equation (8)}
\end{align*}
\]

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**

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

Continue next page
<table>
<thead>
<tr>
<th>Det.</th>
<th>K</th>
<th>block</th>
<th>border</th>
<th># LP iters</th>
<th>Time (s)</th>
<th>Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>6</td>
<td>0.16</td>
<td>0.03</td>
<td>18243</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>a</td>
<td>20</td>
<td>0.05</td>
<td>0.05</td>
<td>46721</td>
<td>160.80</td>
<td>54.30</td>
</tr>
<tr>
<td>a</td>
<td>100</td>
<td>0.01</td>
<td>0.10</td>
<td>1420136</td>
<td>481.10</td>
<td>4784.50</td>
</tr>
<tr>
<td>aC</td>
<td>10</td>
<td>0.27</td>
<td>0.06</td>
<td>276</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>aC</td>
<td>14</td>
<td>0.13</td>
<td>0.07</td>
<td>147856</td>
<td>2853.20</td>
<td>54.40</td>
</tr>
<tr>
<td>aC</td>
<td>22</td>
<td>0.07</td>
<td>0.09</td>
<td>187441</td>
<td>636.70</td>
<td>113.20</td>
</tr>
<tr>
<td>aC</td>
<td>38</td>
<td>0.03</td>
<td>0.11</td>
<td>251646</td>
<td>739.30</td>
<td>190.80</td>
</tr>
<tr>
<td>aC</td>
<td>9</td>
<td>0.32</td>
<td>0.04</td>
<td>273</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>aC</td>
<td>12</td>
<td>0.17</td>
<td>0.12</td>
<td>201955</td>
<td>1415.90</td>
<td>57.20</td>
</tr>
<tr>
<td>aC</td>
<td>16</td>
<td>0.10</td>
<td>0.12</td>
<td>544836</td>
<td>904.40</td>
<td>56.80</td>
</tr>
<tr>
<td>aC</td>
<td>26</td>
<td>0.05</td>
<td>0.14</td>
<td>2553719</td>
<td>2829.10</td>
<td>59.80</td>
</tr>
<tr>
<td>aC</td>
<td>106</td>
<td>0.01</td>
<td>0.18</td>
<td>1589003</td>
<td>1463.90</td>
<td>4784.50</td>
</tr>
<tr>
<td>aCp</td>
<td>10</td>
<td>0.27</td>
<td>0.05</td>
<td>276</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>aCp</td>
<td>22</td>
<td>0.07</td>
<td>0.09</td>
<td>224273</td>
<td>635.90</td>
<td>113.20</td>
</tr>
<tr>
<td>aCp</td>
<td>38</td>
<td>0.03</td>
<td>0.11</td>
<td>265512</td>
<td>727.40</td>
<td>190.80</td>
</tr>
<tr>
<td>aCp</td>
<td>9</td>
<td>0.32</td>
<td>0.04</td>
<td>273</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>aCp</td>
<td>12</td>
<td>0.17</td>
<td>0.10</td>
<td>290977</td>
<td>1039.60</td>
<td>57.20</td>
</tr>
<tr>
<td>aCp</td>
<td>16</td>
<td>0.10</td>
<td>0.11</td>
<td>577623</td>
<td>1019.20</td>
<td>56.60</td>
</tr>
<tr>
<td>aCp</td>
<td>26</td>
<td>0.05</td>
<td>0.12</td>
<td>968633</td>
<td>1266.70</td>
<td>59.70</td>
</tr>
<tr>
<td>aCp</td>
<td>106</td>
<td>0.01</td>
<td>0.16</td>
<td>3211795</td>
<td>3423.60</td>
<td>4784.50</td>
</tr>
<tr>
<td>adC</td>
<td>5</td>
<td>0.27</td>
<td>0.05</td>
<td>271</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>adC</td>
<td>9</td>
<td>0.13</td>
<td>0.07</td>
<td>108036</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>adC</td>
<td>17</td>
<td>0.07</td>
<td>0.09</td>
<td>138008</td>
<td>585.00</td>
<td>113.20</td>
</tr>
<tr>
<td>adC</td>
<td>33</td>
<td>0.03</td>
<td>0.11</td>
<td>206704</td>
<td>609.60</td>
<td>190.80</td>
</tr>
<tr>
<td>adC</td>
<td>4</td>
<td>0.32</td>
<td>0.04</td>
<td>268</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>adC</td>
<td>7</td>
<td>0.17</td>
<td>0.12</td>
<td>285443</td>
<td>1414.80</td>
<td>57.20</td>
</tr>
<tr>
<td>adC</td>
<td>11</td>
<td>0.10</td>
<td>0.12</td>
<td>574388</td>
<td>1030.90</td>
<td>56.80</td>
</tr>
<tr>
<td>adC</td>
<td>21</td>
<td>0.05</td>
<td>0.13</td>
<td>1876435</td>
<td>2067.10</td>
<td>59.80</td>
</tr>
<tr>
<td>adC</td>
<td>101</td>
<td>0.01</td>
<td>0.18</td>
<td>1232117</td>
<td>1703.80</td>
<td>1064.30</td>
</tr>
<tr>
<td>adCp</td>
<td>5</td>
<td>0.27</td>
<td>0.05</td>
<td>271</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>adCp</td>
<td>17</td>
<td>0.07</td>
<td>0.09</td>
<td>154553</td>
<td>543.40</td>
<td>113.20</td>
</tr>
<tr>
<td>adCp</td>
<td>33</td>
<td>0.03</td>
<td>0.11</td>
<td>259472</td>
<td>805.80</td>
<td>190.80</td>
</tr>
<tr>
<td>adCp</td>
<td>4</td>
<td>0.32</td>
<td>0.04</td>
<td>268</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>adCp</td>
<td>7</td>
<td>0.17</td>
<td>0.10</td>
<td>283497</td>
<td>1813.00</td>
<td>57.20</td>
</tr>
<tr>
<td>adCp</td>
<td>11</td>
<td>0.10</td>
<td>0.10</td>
<td>699098</td>
<td>1081.20</td>
<td>56.50</td>
</tr>
<tr>
<td>adCp</td>
<td>21</td>
<td>0.05</td>
<td>0.11</td>
<td>721362</td>
<td>1072.60</td>
<td>59.70</td>
</tr>
<tr>
<td>adCp</td>
<td>101</td>
<td>0.01</td>
<td>0.16</td>
<td>2822062</td>
<td>3518.00</td>
<td>1050.20</td>
</tr>
<tr>
<td>ap</td>
<td>16</td>
<td>0.06</td>
<td>0.04</td>
<td>33182</td>
<td>234.70</td>
<td>54.30</td>
</tr>
<tr>
<td>ap</td>
<td>32</td>
<td>0.03</td>
<td>0.05</td>
<td>44427</td>
<td>99.00</td>
<td>54.30</td>
</tr>
<tr>
<td>ap</td>
<td>10</td>
<td>0.10</td>
<td>0.04</td>
<td>38957</td>
<td>2621.40</td>
<td>54.30</td>
</tr>
<tr>
<td>ap</td>
<td>6</td>
<td>0.16</td>
<td>0.03</td>
<td>23019</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>ap</td>
<td>100</td>
<td>0.01</td>
<td>0.10</td>
<td>326158</td>
<td>385.80</td>
<td>96.70</td>
</tr>
<tr>
<td>apC</td>
<td>38</td>
<td>0.03</td>
<td>0.11</td>
<td>212859</td>
<td>620.80</td>
<td>190.80</td>
</tr>
<tr>
<td>apC</td>
<td>12</td>
<td>0.17</td>
<td>0.11</td>
<td>303335</td>
<td>1840.00</td>
<td>56.90</td>
</tr>
<tr>
<td>apC</td>
<td>16</td>
<td>0.11</td>
<td>0.08</td>
<td>228118</td>
<td>843.70</td>
<td>55.10</td>
</tr>
<tr>
<td>apC</td>
<td>26</td>
<td>0.05</td>
<td>0.12</td>
<td>1179709</td>
<td>1773.90</td>
<td>58.80</td>
</tr>
</tbody>
</table>

Continue next page
<table>
<thead>
<tr>
<th>Det.</th>
<th>K</th>
<th>block</th>
<th>border</th>
<th># LP iters</th>
<th>Time (s)</th>
<th>Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>apC</td>
<td>106</td>
<td>0.01</td>
<td>0.12</td>
<td>1642579</td>
<td>2619.60</td>
<td>910.40</td>
</tr>
<tr>
<td>apCp</td>
<td>38</td>
<td>0.03</td>
<td>0.11</td>
<td>247731</td>
<td>760.40</td>
<td>190.80</td>
</tr>
<tr>
<td>apCp</td>
<td>12</td>
<td>0.17</td>
<td>0.10</td>
<td>332521</td>
<td>1776.50</td>
<td>56.90</td>
</tr>
<tr>
<td>apCp</td>
<td>26</td>
<td>0.05</td>
<td>0.10</td>
<td>57782</td>
<td>1052.80</td>
<td>58.80</td>
</tr>
<tr>
<td>apCp</td>
<td>106</td>
<td>0.01</td>
<td>0.12</td>
<td>1800497</td>
<td>2891.40</td>
<td>910.40</td>
</tr>
<tr>
<td>apdC</td>
<td>33</td>
<td>0.03</td>
<td>0.11</td>
<td>235631</td>
<td>670.50</td>
<td>190.80</td>
</tr>
<tr>
<td>apdC</td>
<td>7</td>
<td>0.17</td>
<td>0.11</td>
<td>330598</td>
<td>1882.80</td>
<td>56.90</td>
</tr>
<tr>
<td>apdC</td>
<td>11</td>
<td>0.11</td>
<td>0.08</td>
<td>348440</td>
<td>836.60</td>
<td>55.10</td>
</tr>
<tr>
<td>apdC</td>
<td>21</td>
<td>0.05</td>
<td>0.11</td>
<td>1327333</td>
<td>1828.00</td>
<td>58.80</td>
</tr>
<tr>
<td>apdC</td>
<td>101</td>
<td>0.01</td>
<td>0.12</td>
<td>740547</td>
<td>1826.10</td>
<td>910.40</td>
</tr>
<tr>
<td>apdCp</td>
<td>7</td>
<td>0.17</td>
<td>0.10</td>
<td>279236</td>
<td>2162.70</td>
<td>56.90</td>
</tr>
<tr>
<td>apdCp</td>
<td>11</td>
<td>0.11</td>
<td>0.08</td>
<td>315679</td>
<td>908.80</td>
<td>55.10</td>
</tr>
<tr>
<td>apdCp</td>
<td>21</td>
<td>0.05</td>
<td>0.10</td>
<td>571431</td>
<td>978.20</td>
<td>58.80</td>
</tr>
<tr>
<td>apdCp</td>
<td>101</td>
<td>0.01</td>
<td>0.12</td>
<td>994508</td>
<td>1950.80</td>
<td>910.40</td>
</tr>
<tr>
<td>cCpCp</td>
<td>24</td>
<td>0.50</td>
<td>0.05</td>
<td>292</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>cCpCp</td>
<td>24</td>
<td>0.49</td>
<td>0.05</td>
<td>902</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>cCpCp</td>
<td>24</td>
<td>0.11</td>
<td>0.15</td>
<td>308</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>cCpCp</td>
<td>24</td>
<td>0.49</td>
<td>0.05</td>
<td>472</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>cCpCp</td>
<td>24</td>
<td>0.11</td>
<td>0.15</td>
<td>322</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>cCpdCp</td>
<td>19</td>
<td>0.10</td>
<td>0.15</td>
<td>392</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>cCpdCp</td>
<td>19</td>
<td>0.10</td>
<td>0.15</td>
<td>1460</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
<tr>
<td>r</td>
<td>3</td>
<td>0.22</td>
<td>0.33</td>
<td>10928</td>
<td>3600.00</td>
<td>4784.50</td>
</tr>
</tbody>
</table>

Table 2: Detailed results for each decomposition

References


20


