A Survey on Decomposition Principles and Methods for the Problem of Railway Traffic Management

Florin Leutwiler [a], Francesco Corman [a,1]
[a] Institute for Transport Planning and Systems, ETH Zurich,
Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland
[1] E-mail: francesco.corman@ivt.baug.ethz.ch, Phone: +41 44 633 33 50

Abstract
Providing punctual, reliable and enough services to customers is one main goal of railway network operators. By automation of train scheduling, it is possible to schedule and route trains on the network closer to its maximal capacity, which is of great value for network operators. In this survey we state the general formulation of the railway scheduling problem and show the principle of decomposition as a way to tackle it. The literature shows many different decomposition approaches. With a survey we aim to summarize existing research and state possible new directions for future research.

Keywords
scheduling, decomposition, benders, dantzig-wolfe, mixed-integer

1 Introduction

Railway traffic management is one fundamental aspect of railway network operation. Providing punctual, reliable and sufficient services to the customers is one main goal of network operators. The operators are keen to route and schedule the movements of trains on the network such that a schedule is robust to smaller disturbance, minimal in operational cost and provides optimal travel times for passengers. Here we identify as schedule both, a plan of operations, such as a timetable, which is planned much before operations, and also its adjustment shortly before operations, or even during the operations, i.e. real time rescheduling. Most of the corresponding scheduling problems are known to be very complex and finding a solution for a problem of practical relevance, e.g. tens of trains and hundreds of railway kilometres, is far from trivial. In consequence a large variety of research has been devoted to the solution of scheduling problems in railway traffic management. A promising approach often seen in the literature is the decomposition of the scheduling problem. For problems with non-deterministic polynomial solution time, it can be highly beneficial to decompose the problem and solve multiple instances of reduced size instead of one large instance to reduce the effect of the scalability issues. In this paper we will state the general formulation of a railway scheduling problem and elaborate the structure within. A survey on principles and methods shall give an overview of well studied decomposition approaches in the literature applicable for scheduling problems and conclude in an advisable direction for future research.

The paper is outlined as follows. In section 2, we state the general problem faced in railway scheduling and provide a brief overview of mathematical formulations of the problem.

In section 3 we explain decomposable structures and how they occurs in scheduling to then provide the basic principles to exploit these structures in the solution process. A overview of practical methods based on principles from section 3 are given in section 4 together with connection to the literature in railway scheduling. The paper concludes in section 5 with a outlook on possible future research.

2 The Problem of Railway Scheduling

The hierarchy in operational planning of railway network operation consists out of three layers according to the time of operations. On the highest layer the line planning is performed to determine necessary connections in between individual stations. On the middle layer the task is to route and schedule individual trains. On the lowest layer the rolling stock planning and crew scheduling is done. In the context of this survey we focus on the middle layer, especially on the scheduling of trains, i.e. below we assume that the route of individual trains are a priori fixed and not open for decision, this happens because either the timetable does not consider rolling stock circulation, as in the typical timetabling problem; or because the rolling stock circulation is already solved, i.e. in the real time rescheduling problem.

2.1 Scheduling

The construction of an operational plan for a railway network with arrival and departure times for every individual train is know as the railway scheduling problem. It inherits decisions on arrival, dwell and departure times as well as ordering of all trains on the network.

Well scheduled operational plans aim to minimize or maximize certain operational aspects, that are crucial for the network operators. The literature provides a large variety of different aspects that are used as objective to scheduling. A very common aspect is the cumulative or largest delay in the network, which operators are keen to keep as small as possible. Alternative aspects, objective to scheduling, seen in the literature are the travel time of passengers, the amount of transported passengers or the operational cost. In reality it is usually a combination of all those aspects that lead to good scheduling.

The scheduling process is always subject to constraints arising from many different sources and affecting the process in different ways. Some constraints have their origin in the non-physical aspects of scheduling, e.g. demands arising from the line planning. Those can be time windows for arrival or departure times of a train, minimal dwell times of a train or connections to other trains on the same station that have to be ensured. The physical side of the railway network restricts the schedule by the dynamics of the specific used rolling stock and operational safety. For safety reasons, trains are not allowed to exceed a certain speed and must follow a minimal distance or time headway in between each other. In addition, no pair of trains can simultaneously use the resource in the network.

The problem of scheduling is to find a detailed schedule of arrival and departure times for all trains at all strategic points on the network. A such schedule must be executable by all trains, must not lead to any conflicts when executed and must not violate any of the operational or physical constraints from above. Possible strategic points on the network are stations, junctions or signals. Preferably the schedule is optimal or near optimal with respect to the objectives important to the network operators.

2.2 Problem formulation

To solve the scheduling problem automatically in an optimal way, it has to be formulated in a well defined mathematical form. The formulation should be able to at least capture all of the crucial objectives and constraints previously mentioned. In the railway scheduling problem considered for this survey, two type of variables are open for decision. First, the order of events happening on the railway network has to be decided, i.e. in which order the trains pass a certain resource of the network. When decided on the order, exact entering and exiting times for all trains and resources in the network have to be decided. In the actual problem only entering or leaving times are necessary, as the time of leaving corresponds to time of entering the next resource. As such, the railway scheduling problem can be easily put in the frame of a job-shop scheduling problem, with train rides corresponding to jobs and resources in the network corresponding to machines. An operation of an individual job then corresponds to a train passing a resource on the network.

In the following we provide a brief overview of different modelling techniques to mathematically capture the railway scheduling problem, common in the literature. The overview is kept brief as the focus will lie on decomposition techniques. For extensive surveys we refer to the surveys by (Fang et al. (2015)), (Corman and Meng (2014)) and (Narayanaswami and Rangaraj (2011)).

Mixed-Integer Linear Program
A mixed-integer linear program is an optimization problem where one part of the optimization variables x are integer, and the remaining variables are continuous variables,

$$\begin{aligned} \min_{x} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b \\ & x \in \mathbb{Z}^n \times \mathbb{R}^m \end{aligned} \tag{1}$$

where A is a matrix and b a vector of appropriate size respectively. With this formulation the scheduling problem can be captured in a very natural way. For tactical decisions such as train ordering and resource assignment, binary variables are introduced in the program. Event times are individually represented by continuous variables. Interactions between events on the railway network are imposed intuitively by linear constraints in the mixed-integer linear program. The literature provides a large variety of mixed-integer formulations for railway (re)scheduling, e.g. (Törnquist and Persson (2005)) and (Tsinghua and Yan (2012)), where a tabu search and commercial solvers respectively are used to retrieve a solution. In (Dessouky et al. (2006)) an alternative mixed-integer formulation is introduced, where a Branch-and-Bound scheme searches the solution space for feasible solutions. For more references on mixed-integer formulations, we refer to (Fang et al. (2015)) and (Narayanaswami and Rangaraj (2011)).

A reduction of the mixed-integer linear program to a pure integer linear program, by removal of the continuous variables is a quiet common alternative to mixed-integer in railway scheduling. Often, integer formulations of scheduling problems are achieved using time discretization. Tactical decisions for scheduling are as in the former case, modelled as binary decision variables. In difference to the mixed-integer formulation, event times are transformed to integer values by time discretization. In (Dollevoet et al. (2012)) an integer programming formulation by time discretization is used for delay management and passenger rerouteing. In a further work it is shown that the same model can be extended to incorporate capacity constraints (Schöbel (2009)). (Caimi et al. (2011)) present an alternative way to formulate the scheduling problem as an integer program. Only a finite set of fixed possible realizations of the continuous variables in a corresponding mixed-integer formulation are considered in the problem. The scheduling problem reduces to a constrained decision process on the different fixed realizations of continuous variables, i.e. an integer linear program. The approach was later used for model predictive rescheduling in (Caimi et al. (2012)). Integer programming formulations are usually applied in cases where a set of solution candidates is simple to generate. For example, integer programming is common in rerouteing problems of railway traffic, where often a set of alternative train routes, provided by heuristics based on the experience of experts, has to be compared and matched to a feasible solution.

Constraint Programming
An alternative to mixed-integer or integer formulations of the scheduling problem is constraint programming. Similar to the former formulations a constraint program bases on a feasible set, defined by a set of constraints, e.g. assignment or disjunctive constrains, which in contrast to mixed-integer programming are mainly logic statements. Typically these are used to assign resources of the network to events or constrain the disjunctive order of events. A slightly different formulation of the constraints in comparison to mixed-integer and integer formulations allows to formulate equivalent problems with fewer variables. Constraint programming does not have an objective such as mixed-integer programming but an evaluation criterion to determine the best solution found. An application of constraints programming is found in (Rodriguez (2007)), where Rodriguez et al. use constraint programming for real-time railway scheduling at junctions.

Alternative Graph
Alternative graphs are a technique introduced by (Mascis and Pacciarelli (2002)) to formulate job-shop scheduling problems and can be used to capture a railway scheduling problem in a special kind of graph theoretical model. In the alternative graph, nodes represent events in time to be scheduled, and arcs represent separation constraints of those events. Two types of arcs exist in the alternative graph. Fixed arcs define an unchangeable relation between events, i.e. a fixed order of events. Alternative arcs occur in pairs, to describe a disjunctive order of two events. A complete selection on an alternative graph, is a selection of alternative arcs, one from each pair, where no additional arc can be selected. Under the condition that the selection represents a meaningful ordering of events, a schedule for the events in the graph is determined by the longest path in the tree of the complete selection of alternative arcs. As show by (Fang et al. (2015)), the alternative graph is a very common formulation technique in railway scheduling, as for example in (Corman et al. (2010)) distributed scheduling of railway systems is performed, based on the alternative graph formulation.

Discrete Event Model
All former formulation techniques inherit all possible orders of trains for every resource of the network. In difference to this, a discrete event model only contains a subset of possible orders. The model is dynamically allocated by forward simulation from a discrete event to the next, where discrete events are basically trains reaching certain strategic points in the

network. For ordering decisions during the forward simulation of the model, commonly heuristics as in (Dorfman and Medanic (2004)) or dynamic programming as in (Ho et al. (1997)) is used. When the forward simulation is terminated, a feasible schedule is found, i.e. the model is dynamically built up during the solution process. (Van den Boom and De Schutter (2006)) and (De Schutter and Van den Boom (2001)) use max-plus switching algebra to formulate out all possible events and use the discrete model in a model predictive control problem solved by a Branch-and-Bound. In (Kersbergen (2015)) a similar model predictive reformulation is used to apply commercial solvers, e.g. CPLEX to the discrete event model formulation.

2.3 Decomposition of Mixed-Integer Linear Programming

In the literature of mathematical programming decomposition is a widely know topic and well established theory exists. In general, special structures within a mathematical program are known to allow a decomposition of the problem, for instance if a variable or a group of variables exists only in a single constraint (typical case is a diagonal matrix). A large variety of different approaches to these structures exists for many different classes of mathematical problems. Decomposition itself is motivated by parallelization, size reduction and simplification of the original problem. Decomposition splits the problem into multiple subproblems of smaller size, which allows for parallel and often more efficient solving, especially when subproblems belong to a class of problems that can be solved very efficiently.

To profit form the theory on decomposition in mathematical programming, we are keen to formulate the railway scheduling problem in an appropriate way. From the models previously introduced, mixed-integer programming is a such appropriate formulation and for the case of railway scheduling the mixed-integer formulation naturally contains the necessary structure to decompose the problem, see section 3. This motivates first a more detailed analysis of the railway scheduling problem structure in a mixed-integer formulation and second a review on approaches for decomposition of mixed-integer programs.

Apart from the benefits of the mixed-integer formulation, there is no big disadvantage caused by restricting ourself to it, as any other of the previously introduced formulations can be transformed into a mixed-integer formulation. Constraint programming can be seen as a complement to mixed-integer linear programming and as mentioned earlier, a constraint program can be casted into a mixed-integer linear program under usage of additional variables. The alternative graph model can be brought in to the form of a mixed-integer linear program by introducing binary decision variables for the pairs of alternatives arcs, together with big-M constraints. Discrete event models can be casted into a mixed-integer program as for example it is done in (Kersbergen (2015)).

In the remaining of the paper we will first elaborate the decomposable structure of the railway scheduling problem formulated as a mixed-integer linear program and present different principles from the literature on decomposition, originated from linear programming with no integer variables in section 3. In section 4 will provide an overview on methods from the literature, that extend the principles form section 3 to the problem of railway scheduling, i.e. a mixed-integer linear program.

3 Decomposable Structure of Scheduling Problems and Principles of Exploration

Generally a mathematical program has to exhibit one out of two possible structure that allow for decomposition, i.e. complicating constraints or complicating variables as described by (Conejo et al. (2005)). The literature shows many different approaches for the decomposition of these two classes of special structured problems. In this section we will first introduce the two structures in case of linear objective and constraints functions as it is the case in a railway scheduling problem and point out the motivation behind the different decomposition approaches, to afterwards show how these structures occur in the railway job-shop scheduling problem. A survey on decomposition principles in linear programming with no integers variables provides a perspective on the basis for today's most common decomposition methods in the complex case of mixed-integer linear programming, reviewed in section 4.

Complicating Constraints

A mathematical program of linear objective and constraints in form of (2) inherit the structure of complicating constraints. A subset of constraints include a large set of the optimization variables of the problem, which prevents a decomposition of the problem.

$$\begin{aligned} \min_{x_1,\cdots,x_K \geq 0} \quad & c_1^\top x_1 + c_2^\top x_2 + \cdots + c_K^\top x_K \\ \text{s.t.} \quad & \begin{bmatrix} A_1 & A_2 & \cdots & A_K \\ B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_K \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix} \geq \begin{bmatrix} b \\ d_1 \\ d_2 \\ \vdots \\ d_K \end{bmatrix} . \end{aligned} \tag{2}$$

The idea of decomposition is based on separating problem (2) into two sub problems, where $x = [x_1, \cdots, x_K]^\top$ and $A = [A_1, \cdots, A_K]$,

$$\begin{aligned} \min_{x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq b \end{aligned} \tag{3}$$

$$\begin{aligned} \min_{x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_K \end{bmatrix} x \geq \begin{bmatrix} d_1 \\ \vdots \\ d_K \end{bmatrix} . \end{aligned} \tag{4}$$

In the literature (3) is usually called master problem, while (4) is referred as subproblem. It easy to see that solving (3) and (4) does not necessarily provide a solution to (2). We will later see appropriate modifications of the problems (3) and (4) to turn them into an equivalent representation of (2). Note that by its diagonal structure, problem (4) can be decomposed into K individual problems, which can be solved independently. Principles in decomposition of complicating constraints are keen to not destroy the diagonal structure through subproblem modifications. In addition, the subproblems are often significantly simpler than the original problem, what together with the separation of subproblems motivated many principles and methods in complicating constraints, reviewed in section 3.2 and 4.1 respectively.

Complicating Variables

The complementary case to complicating constrains in a mathematical program with linear objective and constraints, are complicating variables. A small subset of optimization variables is present in a large set of constraints, as in (5).

$$\begin{aligned}\min_{y,x_1,\cdots,x_K\geq 0} \quad & c_0^\top y + c_1^\top x_1 + c_2^\top x_2 + \cdots + c_K^\top x_K \\ \text{s.t.} \quad & \begin{bmatrix} A_1 & B_1 & & \\ A_2 & & B_2 & \\ \vdots & & & \ddots & \\ A_K & & & & B_K \end{bmatrix} \begin{bmatrix} y \\ x_1 \\ \vdots \\ x_K \end{bmatrix} \geq \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_K \end{bmatrix}.\end{aligned} \tag{5}$$

The decomposition approach is similar to the case of complicating variables. The problem is separated into two problems by individually optimize over y and x, with $A = [A_1, \cdots, A_K]^\top$ and $x = [x_1, \cdots, x_K]^\top$,

$$\begin{aligned}\min_{y\geq 0} \quad & c^\top y \\ \text{s.t.} \quad & Ay \geq \begin{bmatrix} d_1 \\ \vdots \\ d_K \end{bmatrix}\end{aligned} \tag{6}$$

$$\begin{aligned}\min_{x\geq 0} \quad & c^\top x \\ \text{s.t.} \quad & \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_K \end{bmatrix} x \geq \begin{bmatrix} d_1 \\ \vdots \\ d_K \end{bmatrix}.\end{aligned} \tag{7}$$

The master (6) and subproblem (7) are not yet an exact equivalent to (5) and similar to complicating constraints, modification to both problems have to be made. Note, again the subproblem (7) has diagonal structure and can be nicely decomposed, as long as the necessary subproblem modifications do not destroy the structure. This observation, together with the fact that the subproblem is often significantly simpler than the original problem motivated principles and methods for complicating variables, reviewed in section 3.3 and 4.2 respectively.

3.1 Structure of Railway Scheduling Problems

In the railway scheduling problem both types of decomposable structures naturally occur. Depending on how the optimization variables are grouped, one of the structures establishes. In section 2 we showed that two type of variables, i.e. event times and ordering decisions are open for decision, in the railway job-shop scheduling problem as we consider it in this survey. Recall that in the problem, train rides refer to jobs, network resources refer to machines, and a train passing a resource is referred to an operation of the corresponding job. Defining these variables by $t_{i,j,m}$ for the event times and $p_{i,j,m}$ for the decisions, each variable has three indices,

$$t_{i,j,m},\, p_{i,j,m}: \begin{cases} i & \text{operation index} \\ j & \text{job index} \\ m & \text{machine index} \end{cases}.$$

In accordance with the indices of the variables, we can group either the variables or the constraints of the scheduling problem by three criteria i.e, by groups of machines, by groups of jobs or by groups of operations. Grouping by operations refers to grouping variables by the assumed event time of the associated operation into groups of different time windows. Figure 1 illustrates on the constraint matrix how grouping by variables lead to the structure of complicating constraints, while grouping by constrains lead to complicating variables

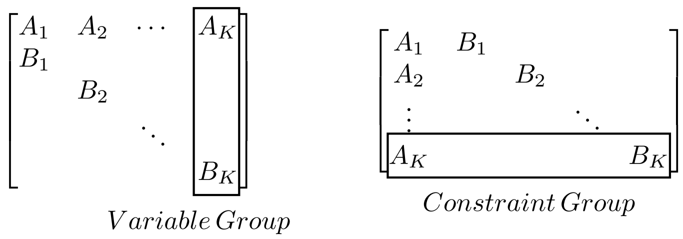

Figure 1: Grouping of variables: Complicating Constraints (left), Variables (right).

The grouping with respect to resources has a natural intuition for variable and constraint grouping. When grouping variables, this corresponds to a separation of the railway network in equivalent regions, all connected to some neighbours, see Figure 2. In (Caimi et al. (2009a)), a such separation is used for a decomposition approach. When constraints are grouped, the constraint matrix turns out to have the same structure as when the network is decomposed in a master/slave manner, where sub regions of the network are slaves only connected with the master region, see Figure 2. A similar decomposition is applied in (Lamorgese and Mannino (2015)).

The variable grouping with respect to operations is similar to what is often done in temporal decomposition of the railway scheduling problem, e.g. the rolling horizon approach. Variables are grouped by time windows in which the according events are expected to occur.

As mentioned earlier, the natural occurrence of the decomposable structure is what motivates a review of decomposition principles in linear programming with no integer variables, i.e. the basis of decomposition methods in mixed-integer linear programming, which help to tackle the problem of railway scheduling.

3.2 Principles for Complicating Constraints

For the structural exploration of a linear program with no integer variables and complicating constraints, there are two very common principles in the literature. Both are motivated by the idea of separating the complicating constraints from the diagonally structured part of the problem, to make it simpler and decomposable.

Dantzig-Wolfe Reformulation

The reformulation of Dantzig-Wolfe is based on an alternative representation of a polytope. A polytope, such as those in the diagonally structured part of problem (2),

$$x_k \in P_{B,k} := \{x \in \mathbb{R}_+ : \; B_k x \geq d_k\} \tag{8}$$

can be represented by a convex combination of its extreme points and extreme rays,

$$x_k = \sum_{v_g \in G_k} \lambda_g v_g + \sum_{w_q \in Q_k} \theta_q w_q, \quad \text{where} \sum_{v_g \in G_k} \lambda_g = 1, \; \lambda_g, \theta_q \geq 0 \tag{9}$$

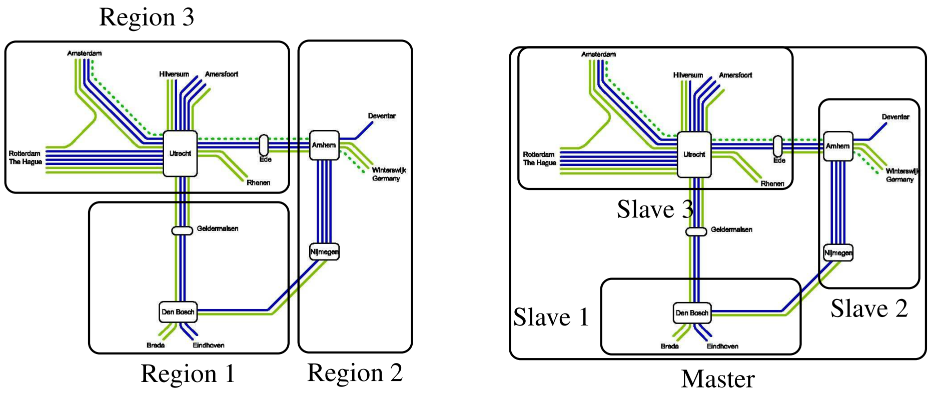

Figure 2: Decentralized decomposition $\{l.\}$, Master/Slave decomposition $\{r.\}$

where v_g, $g \in G_k$ and w_q, $q \in Q_k$ are extreme points and rays of $P_{B,k}$ respectively, a proof is given in (Bertsimas and Tsitsiklis (1997)). For the readability we assume the polytopes $P_{B,k}$ to be bounded , i.e. $Q_k = \emptyset$. The general case is straight forward and can be found in (Bertsimas and Tsitsiklis (1997)). With the alternative formulation of a polytope, linear problem (2) can be reformulated to,

$$\begin{aligned} \min_{\lambda} \quad & \sum_{K} \sum_{v_g \in G_k} \left(c_k^\top v_g\right) \lambda_g \\ \text{s.t.} \quad & \sum_{K} \sum_{v_g \in G_k} \left(A_k v_g\right) \lambda_g \geq b \\ & \sum_{g \in G_k} \lambda_g = 1, \, \lambda_g \geq 0 \qquad \forall k \in K \end{aligned} \tag{10}$$

which is known as Dantzig-Wolfe reformulation. The cardinality of the sets G_k can be exponentially large and it would be computationally very costly to perform the optimization in (10) over all elements of G_k. The method of column generation allows to perform the optimization in (10) over only a reduced subset of G_k, i.e. $\tilde{G}_k$. Optimal dual multipliers of the reduced problem provide information on how relaxation or tightening of an individual constraint affect the objective value. These multipliers are used to search for elements in G_k, that reduce the objective value and therefore must be added to $\tilde{G}_k$ to achieve an optimal solution of (10). The search is known as pricing problem in column generation and, by the diagonal structure in problem (2), can be performed over the individual polytopes corresponding to G_k separately,

$$\begin{aligned} \min_{x_k} \quad & \left(c_k^\top - u^\top A_k\right) x_k \\ \text{s.t.} \quad & x_k \in P_{B,k} \end{aligned} \tag{11}$$

where u are the optimal dual multipliers of (10). In case of (11) the optimal solution x_k^* is either an extreme point or ray of $P_{B,k}$, i.e. $x_k^* \in G_k$. In cases of strong duality, e.g. linear programming with no integer variables, where the primal and dual are equivalent, column generation is guaranteed to find an optimal solution, if one exists. Similar Dantzig-Wolfe reformulations exist for linear programming with presence of integer variables, as it is the case railway scheduling. Though one has to be aware that column generation in presence of integer variables does no longer find an optimal solution but only provide a primal bound on the optimization problem, as a result of the absence of strong duality. To solve the original mixed-integer linear program additional techniques are necessary, which will be reviewed in section 4. (Vanderbeck and Savelsbergh (2006)) propose an extension of the Dantzig-Wolfe reformulation to the case of mixed-integer linear programming, in form of two different approaches, i.e. convexification and discretization. In the convexification approach, $P_{B,k}$ in (11) is replaced by the generating set G_k of its convex hull,

$$G_k = \{(x_g, y_g) \in \mathbb{R}^n \times \mathbb{Z}^p : (x_g, y_g) = \text{extreme points of } conv(P_{B,k})\}\,. \tag{12}$$

To impose integraliy in the convexification approach an additional integrality constraint is necessary in (10),

$$\sum_{y_g \in G_k} y_g \lambda_g \in \mathbb{Z}^p. \tag{13}$$

The discretization approach in (Vanderbeck and Savelsbergh (2006)) is originated in integer polytope theory. A rational integer polytope is known to be representable by a combination of finitely many integer extreme points and rays of the polytope, similar to (9). In case of mixed-integer, integer variables are then represented by the finite integer set, while continuous variables are reformulated using the same technique as in the convexification approach. We refer to (Vanderbeck and Savelsbergh (2006)) for exact definition of the generating integer and continuous variable sets. In the discretization approach, the constraint (13) can be replaced by the restriction $\lambda_g \in \{0,1\}$. The corresponding pricing problems in both approaches are identical to (11) except $x \in G_k$ and no longer in $P_{B,k}$. Note that in case of binary integer variables in the original problem (2) convexification and discretization are identical, as no integer interior points exists in $P_{B,k}$. (Vanderbeck and Savelsbergh (2006)) provide further techniques to relax and simplify the generating sets of the discretization approach. Also Vanderbeck presents in (Vanderbeck (2000)) further alternative reformulations of the generating sets in Dantzig-Wolfe decomposition.

Lagrangian Relaxation

The Lagrangian relaxation is less commonly used in decomposition as it is the case for Dantzig-Wolfe reformulation. More often Lagrangian relaxation is used for dual bounds on the objective value. Nontheless it can be used for decomposition in linear programming with no integer variables and complicating constraints as in (Conejo et al. (2005)). The idea is to relax the complicating constraints into the objective, such that the optimization problem becomes decomposable,

$$\begin{aligned} z(u) = \min_{x_1, x_2, \cdots, x_K} u^\top b + \sum_K \left(c_k^\top - u^\top A_k\right) x_k \\ \text{s.t.} \begin{bmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_K \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix} \geq \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_K . \end{bmatrix} \end{aligned} \tag{14}$$

Note that the Lagrangian relaxation reformulation is the dual of the Dantzig-Wolfe reformulation. As show in (Wolsey and Vanderbeck (2010)) the dual problem of (2) is given by maximizing Lagrangian relaxation (14) over u,

$$\begin{aligned} \max_{u} \quad & u^\top b + \sigma \\ \text{s.t.} \quad & u^\top A_k v_g + \sigma \leq c^\top v_g \quad \forall g \in G_k,\ k \in [K]. \end{aligned} \tag{15}$$

The optimization over x in (14) has been replaced by the constraint in (15). The dual of (15) is exactly (10). In cases of strong duality, solving dual problem $z_D = \max_u z(u)$ from relaxation of the complicating constraints results in a optimal solution pair of primal and dual variables (x^*, u^*). The dual problem z_D is commonly solved using a scaled gradient method for the optimization over dual variables u as in (Fisher (1981)), while the optimization over the primal variables x_k is done explicitly, where the diagonal structure permits a decomposed optimization. For the Lagrangian dual of mixed-integer programs, strong duality does not hold in most cases. To this topic, (Geoffrion (1974)) provides deeper insights on the Lagrangian bound for integer programming and in which cases, strong duality still holds. In section 4 we present techniques that will help to overcome the lack of strong duality, to solve problem (14) to integer optimality.

3.3 Principles for Complicating Variables

For the structural exploration in linear programming with no integer variables and complicating variables, a large variety of different principles exist in the literature. Motivated are these approaches through the observation, that fixing the complicating variables in a problem of complicating constraints turns the problem into a significantly simpler subproblem, in case of problem (5) even into a diagonally structured subproblem. The idea is to separate the optimization over the complicating variables from the optimization over the non-complicating variables in which complicating variables are fixed. Note, a solution to the complicating variables is an optimal solution to the original problem, if also a solution to the corresponding subproblem exists. If no solution to the subproblem exists, different principles generate different types of constraints, i.e. cuts for the optimization over the complicating variables, depending on the feasibility and optimality of the subproblem. In the following we present the most common principle in the literature, which lead to numerous abbreviated methods in mixed-integer linear programming for solving the case of complicating constraints, reviewed in section 4.

Benders Decomposition
One of the earliest and most common cut generating principle in linear programming is the decomposition method introduced by (Benders (1962)). By the dual representation of the subproblem feasible set, feasibility cuts are generated. In addition, the dual representation of the subproblem provides a lower bound on the contribution of non-complicating variables to the objective of the original problem, that can be used as an optimality cut. In case of problem (5), fixing the complicating variables y together with the diagonal structure yields K different subproblems,

$$\begin{aligned} \min_{x_k \geq 0} \quad & c_k^\top x_k \\ \text{s.t} \quad & B_k x_k \geq d_k - A_k y \end{aligned} \tag{16}$$

and the K different dual subproblems respectively,

$$\begin{aligned} \max_{u_k \geq 0} \quad & u_k^\top (d_k - A_k y) \\ \text{s.t.} \quad & u_k^\top B_k \leq c_k. \end{aligned} \tag{17}$$

In the iterative process of Benders decomposition the master problem (5) is relaxed to possible constraints on y and from iteration to iteration more cuts are added to this master problem. Given an optimal solution (y_i^*, x_i^*) to the relaxed master problem at the i-th iteration, the following three cases as described in (Bertsimas and Tsitsiklis (1997)) may occur when solving the dual subproblem (17),

Case I: The dual subproblem has a feasible optimal solution $u_{k,i}^*$. If the inequality $u_{k,i}^{*\top}(d_k - A_k y_i) > c_k^\top x_{k,i}^*$ holds, the primal solution $x_{k,i}^*$ of the master problem is infeasible to the primal subproblem and the cut must be added to the master problem,

$$u_k^{*\top}(d_k - A_k y) \leq c_k^\top x_k. \tag{18}$$

Case II: The dual subproblem is unbounded, implying the primal subproblem is generally infeasible. In this case, there exists an extreme ray $w_{k,i}$ with $w_{k,i}(d_k - A_k y_i) > 0$, which causes the unboundedness of the dual subproblem and has to be excluded by the following cut in the master problem,

$$w_{k,i}(d_k - A_k y_i) \leq 0 \tag{19}$$

Case III: All dual subproblems have a feasible solution and the master solution $x_{k,i}^*$ is primal subproblem feasible, i.e. $u_{k,i}^{*\top}(d_k - A_k y_i) \leq c_k^\top x_{k,i}^*$ for all $k \in [K]$. In this case (y_i^*, x_i^*) is an optimal solution to (5).

Note that in the case where the subproblems are linear and contain no integer variables, the Benders cuts (i) and (ii) are valid, i.e. they are guaranteed to cut of part of the solution space. This property is usually lost with the presence of integer variables in the subproblem. Devoted to this problem, the literature holds a variety of generalizations and techniques to overcome this issue. At first (Geoffrion (1972)) generalized the Benders decomposition to a more general class of problems of the form,

$$\begin{aligned} \min_{y,x} \quad & f(y,x) \\ \text{s.t.} \quad & G(y,x) \geq 0 \end{aligned} \tag{20}$$

where (20) does not have to be linear or even convex over y and x jointly but only convex over x when complicating variables y are fixed. A further generalization to non-convex subproblems, e.g. mixed-integer linear subproblems is given in (Wolsey (1981)). But as mentioned by Wolsey the optimization in the subproblems is usually not practically possible, only in special cases, e.g. the case of a convex subproblem investigated by Geoffrion.

A different direction of generalization of Benders decomposition is introduced by (Hooker and Ottosson (2003)) through logic-based Benders decomposition, where they introduce the inference dual of an optimization problem for the generation of different Benders cuts. The inference dual is a generalized concept of duality, which does hold for mixed-integer linear subproblems. In (Hooker and Ottosson (2003)) an example on a multi-machine scheduling problem is provided, where the inference duality concept generates cuts, which exclude a infeasible set of job assignments to one machine, in the master problem.

In section 4 we will show methods from the literature, based on the former principles of Benders decomposition to overcome the lack of strong duality and generate valid cuts in the case of mixed-integer linear programming, e.g. railway scheduling.

4 Practical Decomposition Methods

The principles in section 3 are mostly only valid for linear / convex problems and exhaustively make use of duality theory, especially the theorem of strong duality. The extension of these principles to classes of problems where strong duality does not hold, e.g. mixed-integer linear programming is usually not straight forward and often theoretical concepts on this topic are not practical for actual computations. Still, the literature contains many different methods, proposing practical algorithms based on principles from section 3.

4.1 Methods for Complicating Constraints

In case of missing strong duality, e.g. the mixed-integer linear program formulation of railway scheduling, the Dantzig-Wolfe and Lagrangian reformulation no longer provide optimal solutions, but only bounds on the objective of an optimization problem. In practical methods, the use of these reformulation is motivated by the achievement of better bounds on the objective value, often of great use in the solution process. For example in branching schemes, tighter bounds can cause significant increase in performance of the overall method. Furthermore, the computation of bounds becomes more efficient, thanks to decomposition by reformulation.

Methods on Dantzig-Wolfe Decomposition

Dantzig-Wolfe reformulation allows to decompose problems of complicating constraints by a reformulation based on the alternative representation of the feasible subproblem sets. Column generation prevents the reformulation from becoming to large to solve. To retrieve integer solutions in a Dantzig-Wolfe column generation reformulation, branching is necessary, (Desaulniers et al. (2005)). The branching scheme imposes integrality on the relaxed reformulation, where relaxation is necessary to apply column generation. These methods are known as Branch-and-Price, i.e. a branching tree is built up, where at each node column generation is performed upto (sub)optimality and then a fractional part of solution is branched. Branch-and-Price inherits several practical issues that have to be taken care of. Column generation has to be performed at every node as it is possible that a column, non-optimal at an earlier node becomes cost reducing on the current node, due to additional integrality restrictions from branching. Considering branching itself, it is known for a fact that standard branching schemes are not suitable for an integer program from a Dantzig-Wolfe reformulation, (Vanderbeck (2000)). Alternative branching schemes have been proposed in (Vanderbeck and Wolsey (1996)), (Barnhart et al. (1998)) and in more general form in (Vanderbeck (2000)). Instead of branching on individual λ_g variables in problem (10), a common alternative is to branch on the sum of a subset of λ_g variables,

$$\sum_{g \in \bar{G} \subseteq \tilde{G}} \lambda_g \geq \lceil \alpha \rceil \, ; \qquad \sum_{g \in \bar{G} \subseteq \tilde{G}} \lambda_g \leq \lfloor \alpha \rfloor \tag{21}$$

for the upper and lower branch respectively. The constant $\alpha = \sum_{g \in \bar{G} \subseteq \tilde{G}} \lambda_g$ is fractional, where, e.g. in (Vanderbeck and Wolsey (1996)) it has been proven that a such subset $\bar{G}$ exists, if a solution λ^* to (10) is fractional. An other alternative to standard branching is to branch on the original variables of problem (2), which is applied in case of different subproblems,

$$\sum_{g \in \bar{G}_k \subseteq \tilde{G}_k} v_g \lambda_g \geq \lceil \alpha_k \rceil \, ; \qquad \sum_{g \in \bar{G}_k \subseteq \tilde{G}_k} v_g \lambda_g \leq \lfloor \alpha_k \rfloor \tag{22}$$

with α_k similarly defined as before. The branching strategy (22) can either be implemented in the master problem or in the individual subproblems as in (Wolsey and Vanderbeck (2010)). Branching in subproblems can be favourable, as it imposes tighter bounds, but in contrary results in sometimes much more complex subproblems. Branch-and-Price is often initialized by a heuristic to determine a first version of the generating set $\tilde{G}_k$. More insight on initialization is given in (Vanderbeck (2005)). Early termination of the column generation at the nodes is discussed in (Vanderbeck and Wolsey (1996)).

Branch-and-Price methods are often used in routing problems in the areas of railway (Peeters and Kroon (2008)), air (Sarac et al. (2006)) or vehicle traffic (Dabia et al. (2013)) or crew scheduling problems. In these problems, the subproblems to be solved during the Branch-and-Price procedure usually have special structure, which allows them to be solved by fast algorithms, such as dynamic programming, e.g. in (Caprara et al. (2003)). In (Chu (2018)) Branch-and-Price with additional generation of cuts is applied to design and simultaneously plan a schedule for an urban bus network. More common in the railway scheduling literature is column generation without branching as a heuristic, e.g. in (Toletti (2018)). Similar to Branch-and-Price, decomposition is also possible for these heuristic approaches but optimality is no longer guaranteed.

Methods on Lagrangian Relaxation

Also for Lagrangian relaxation, the presence of integer variables, in most cases, prohibit strong duality and the relaxed formulation provides only a lower bound on the objective of (2). Similar to the case of Dantzig-Wolfe reformulation, the retrieved lower bounds are used in branching schemes, generally known as Branch-and-Bound. Inside the scheme, at each node a lower bound is retrieved trough the Lagrangian relaxation. The bound then can be used to determine if the subtree rooted at the current node can be fathomed or not. The optimization over x variables with u variables fixed, is often carried out explicitly, as the relaxed problem is usually significantly simpler to solve and in case of problem (5), by the diagonal structure, even decomposable. For the optimization over u, approximative methods, e.g. the sub gradient method, are used, e.g. in (Fisher (1981)). Optimal dual multipliers u^* are found when 0 is a subgradient of $z(u^*)$.

Branch-and-Bound using Lagrangian relaxation is very popular in many different fields including scheduling of railway networks, see for example (Zhou and Zhong (2007)) or (Brännlund et al. (1998)). Zhou and Zhong presented a Branch-and-Bound algorithm for single-track scheduling using Lagrangian relaxation to retrieve good lower bounds.

4.2 Methods for Complicating Variables

Decomposition techniques in presence of complicating variables are motivated by the generation of constraints, i.e. cuts for the master part problem from subproblems. Whenever these cuts base on duality theory, i.e. strong duality, they can become redundant and do not guarantee progress in the solution process any more when integer variables are present. In the literature a variety of techniques for the decomposition of mixed-integer problems with complicating variables exists, where modification allow to find valid cuts despite the presence of integer variables.

Methods on Benders Decomposition

In Benders decomposition the dual objective of individual subproblems is used to generate restrictions on the master problem. By the possible duality gap in presence of integer variables in problem (2), constraints from the dual objective can become redundant and therefore useless. Still Benders decomposition is commonly used in mixed-integer linear programming, but under a slightly different definition of the complicating variables as introduced in section 3. In the following we present known methods for the alternative complicating variable definition, to afterwards present ideas from the literature to handle the original definition of complicating variables from section 3.

Standard applications of Benders decomposition in mixed-integer linear programming aim to separate integer variables as complicating variables form continuous variables. As a result, no integer variables are present in any subproblem, wherefore the cuts remain valid and if the feasible set of integer solutions in the master problem is finite, termination of the iterative cutting process is given. When separating integer variables from continuous in a railway scheduling problem, the possibility of a proper diagonal structure in the subproblem as in section 3 is very unlikely. Often in a larger railway network chains of interactions through the entire network exist, i.e. every resource and every train, through other trains affect almost any other train in the network and thus no diagonal structure establishes. Still, the approach recently has been used successfully for railway scheduling in (Lamorgese and Mannino (2019)), where detailed analysis of the subproblem revealed an alternative decomposition technique in contrast to the diagonal structure.

Instead of solving the master problem at each iteration of the cut generation process to integer optimality, cutting can be included into a branching scheme as in (Naoum-Sawaya and Elhedhli (2013)), a Branch-and-Cut methods is the result. Benders Branch-and-Cut only solves a relaxation with respect to integrality of master problem at each node. After Benders cuts are generated at the individual nodes, ideally until no further cuts can be found, branching is performed on a fractional part of the solution. Note, that also in case of Benders Branch-and-Cut, integer variables have to be separated from continuous to generate continuous subproblems, i.e. nothing can be gained form the diagonal structure in (5).

Different to the standard Benders approach in mixed-integer linear programming are logic Benders based approaches. Again integer variables are separated from continuous, but cuts are based on logic statements instead of Lagrangian duality. (Codato and Fischetti (2006)) introduce combinatorial Benders cuts to separate sets of disjunctive relations on continuous variables from the master problem. Binary integer variables in the master problem define which of the disjunctive constraints are active. If the subproblem, i.e. the set of active disjunctive constraints turns out to be infeasible, a cut removes an irreducible infeasible set of selected active disjunctive constraints C from the master problem in form of a constraint on the binary integer variables in the master problem,

$$\sum_{i \in C;\, y_i = 0} y_i + \sum_{i \in C;\, y_i = 1} (1 - y_i) \geq 1. \tag{23}$$

A similar scheme is used in the hybrid logic Benders approach presented in (Harjunkoski et al. (2000)) or (Harjunkoski and Grossmann (2002)). In difference to combinatorial Benders cuts, no irreducible infeasible set is determined, instead constraint programming is used to determine the feasibility of a subproblem. Integer cuts similat to (23) are added to the master in case of infeasible subproblems. (Harjunkoski and Grossmann (2002)) provide an application of the hybrid approach on multi-machine scheduling. In (Lamorgese and Mannino (2015)) and (Lamorgese et al. (2016)) a similar approach is used for railway scheduling, where the platform assignment problem at stations is decoupled from the remaining problem through a Benders like reformulation.

Regarding Benders decomposition for exploration of the diagonal structure in the railway scheduling of form (5), some approaches can be found in the literature to handle mixed-integer linear subproblems. (Chu and Xia (2004)) present a modification of the subproblem such that a Benders cut can be generated, that is proven to cut off at least one integer solution point in the master, such that by finiteness of integer points in the master, termination is guaranteed. Other approaches try to find the convex hull of the feasible subproblem set to make the integer restrictions redundant and return to a subproblem without integer variables, as in the former cases. (Sherali and Fraticelli (2002)) use sequential lift-and-project from integer programming to remove fractional variables in the optimal subproblem solution. (Sen and Sherali (2006)) present an alternative approach solving the subproblems through Branch-and-Cut, where in after termination the cuts construct the facets of the convex hull.

4.3 Heuristic Decomposition

In difference to exact decomposition techniques, heuristics are very common in the railway scheduling literature. Here we briefly mention some heuristics, encountered during the review of literature.

(Corman et al. (2010)) present a geographically decomposed scheduling approach. Local scheduling on different regions is globally aligned by an iterative heuristic. The heuristic has been proven to, if successful, return global feasible schedules. Optimality nontheless is not guaranteed. The approach is a heuristic solution technique to the railway scheduling formulation with complicating constraints as presented in section 3.

Temporal decomposition is an other common approach handled by heuristics in the literature. Decomposition in time propose a complicating constraint formulation of the problem. Rolling horizon approaches are frequently used to handle such problems. In rolling horizon, subproblems over fragments of the time horizon are sequentially solved, beginning with the first fragment in time and ending with the last. In between fragments, variables shared with the previous fragment are assumed to be partially or fully fixed. (Tsinghua and Yan (2012)) successfully apply a rolling horizon approach to railway scheduling.

An different approach in (Caimi et al. (2009b)) separates the problem inspired by its geographic structure into condensation and compensations zones, i.e. a distributed geographic decomposition. Caimi et al. noted that railway networks often explore zones of high density and opposite zones of low density. It is likely that giving precedence in scheduling to condensation zones and then schedule compensations, with fixed boundary conditions for condensation zones, compensations zones are able to compensate delays from their neighbouring zones. The scheduling of condensation and compensation is alternated, until a feasible schedule is found.

5 Discussion

Decomposition in railway scheduling, i.e. the planning of arrival, departure and passing times at strategical points for each train operating on a railway network has received more and more attention over the past years. In this paper we present a brief overview on formulations of the railway scheduling problem, to then focus on the mixed-integer linear programming formulation of a railway scheduling problem. A natural occurrence of decomposable structures in the railway scheduling problem is shown and corresponding interpretations of the structures are given. The literature contains a variety of general approaches to handle decomposable structures efficiently. By making the link to the railway literature it can be seen that some of these approaches are already successfully in use for railway scheduling, especially the methods of Branch-and-Price (Dantzig-Wolfe) as well as Lagrangian relaxation. On the other side, methods such as Benders decomposition are less common in the railway literature. Most often in the literature of railway scheduling, whenever decomposition is the goal, specifically designed heuristics are used. In many cases, these are closely related to decomposition principles in linear programming, some of which are shown in section 3, e.g. column generation for Dantzig-Wolfe like reformulations.

Decomposition will continue to be one major topic in the future research of railway scheduling as railway operators and researchers continuously aim to tackle larger scenarios.

On one side the increase in size will pose more and more complicated mathematical problems, which have to be tackled in the right way. For future directions of research on how to tackle such complex mathematical problems, it can be potentially interesting to explore the full potential of the Benders decomposition. Compared to methods in Branch-and-Price (Dantzig-Wolfe), Bender decomposition has received much less attention in railway scheduling. In comparison to other research fields, e.g. stochastic optimization, which is also a problem in railway scheduling, Benders decomposition has received large attention and many successful applications exist. Especially approaches that are able to handle mixed-integer linear subproblems can be of great value to railway scheduling as they allow a very direct and natural decomposition of large scale railway scheduling problems.

On the other side, not only pose large scale problem in railway scheduling significantly more complex mathematical problems but also demand larger amount of computational power and memory. Decomposition can not only help to simplify the problem mathematically, but also allow to split up the computation onto multiple smaller sized machines, resulting in possibly lower cost for the computational infrastructure.

In the future of railway operation, decomposition can be one key component to the automation of scheduling operations on large scale railway networks for entire countries in an optimal manner.

Acknowledgements

This work was supported by the SBB and ETH Zürich Foundation.

References

C. Barnhart, E. L. Johnson, G. L. Nernhauser, M. W. P. Savelsbergh, P. H. Vance, and G. L. Nemhauser. Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research*, 46(3):316–329, 1998.

J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

D. J. Bertsimas and J. J. N. J. Tsitsiklis. *Introduction to Linear Optimization*. 1997.

U. Brännlund, P. O. Lindberg, A. Nõu, and J.-E. Nilsson. Railway Timetabling Using Lagrangian Relaxation. *Transportation Science*, 32(4):358–369, 1998.

G. Caimi, D. Burkolter, T. Herrmann, F. Chudak, and M. Laumanns. Design of a railway scheduling model for dense services. In *Networks and Spatial Economics*, volume 9, pages 25–46, 2009a.

G. Caimi, M. Fuchsberger, D. Burkolter, T. Herrmann, R. Wüst, and S. Roos. Conflict-free train scheduling in a compensation zone exploiting the speed profile. *International Seminar on Railway Operations Research*, (January):162–177, 2009b.

G. Caimi, F. Chudak, M. Fuchsberger, M. Laumanns, and R. Zenklusen. A New Resource-Constrained Multicommodity Flow Model for Conflict-Free Train Routing and Scheduling. *Transportation Science*, 45(2):212–227, 2011.

G. Caimi, M. Fuchsberger, M. Laumanns, and M. Lüthi. A model predictive control approach for discrete-time rescheduling in complex central railway station areas. *Computers and Operations Research*, 39(11):2578–2593, 2012.

A. Caprara, M. Monaci, and P. Toth. Models and algorithms for a staff scheduling problem. *Mathematical Programming*, 98:445–476, 2003.

J. C. Chu. Mixed-integer programming model and branch-and-price-and-cut algorithm for urban bus network design and timetabling. *Transportation Research Part B: Methodological*, 108:188–216, 2018.

Y. Chu and Q. Xia. Generating Benders cuts for a general class of integer programming problems. *First International Conference, CPAIOR*, pages 127–141, 2004.

G. Codato and M. Fischetti. Combinatorial Benders' Cuts for Mixed-Integer Linear Programming. *Operations Research*, 54(4):756–766, 2006.

A. Conejo, E. Castillo, R. Mínguez, and R. García-Bertrand. *Decomposition Techniques in Mathematical Programming*. Springer, 2005.

F. Corman and L. Meng. A Review of Online Dynamic Models and Algorithms for Railway Traffic Management. *IEEE Transactions on Intelligent Transportation Systems*, 9(3): 1–11, 2014.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Centralized versus distributed systems to reschedule trains in two dispatching areas. *Public Transport*, 2(3):219–247, 2010.

S. Dabia, S. Ropke, T. van Woensel, and T. De Kok. Branch and Price for the Time-Dependent Vehicle Routing Problem with Time Windows. *Transportation Science*, 47 (3):380–396, 2013.

B. De Schutter and T. Van den Boom. Model predictive control for railway networks. *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No.01TH8556)*, 1:105–110, 2001.

G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Column Generation*. Springer, 2005.

M. M. Dessouky, Q. Lu, J. Zhao, and R. C. Leachman. An exact solution procedure to determine the optimal dispatching times for complex rail networks. *IIE Transactions (Institute of Industrial Engineers)*, 38(2):141–152, 2006.

T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay Management with Rerouting of Passengers. *Transportation Science*, 46(1):74–89, 2012.

M. J. Dorfman and J. Medanic. Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Research Part B: Methodological*, 38(1): 81–98, 2004.

W. Fang, S. Yang, and X. Yao. A Survey on Problem Models and Solution Approaches to Rescheduling in Railway Networks. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2997–3016, 2015.

M. L. Fisher. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science*, 27(1):1–18, 1981.

A. Geoffrion. Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

A. Geoffrion. Lagrangean Relaxation for Integer Programming. *Mathematical Programming Study*, 2(2):82–114, 1974.

I. Harjunkoski and I. E. Grossmann. Decomposition Techniques for Multistage Scheduling Problems using MIP and CP Methods. *Computers and Chemical Engineering*, 26(11): 1533–1552, 2002.

I. Harjunkoski, V. Jain, and I. E. Grossmann. Hybrid mixed-integer/constraint logic programming strategies for solving scheduling and combinatorial optimization problems. *Computers and Chemical Engineering*, 24(2-7):337–343, 2000.

T. K. Ho, J. P. Norton, and C. J. Goodman. Optimal traffic control at railway junctions. *IEE Proceedings - Electric Power Applications*, 144(2):140–148, 1997.

J. N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming, Series A*, 96(November 2000):33–60, 2003.

B. Kersbergen. *Modeling and Control of Switching Max-Plus-Linear Systems: Rescheduling of railway traffic and changing gaits in legged locomotion*. Phd thesis, Technische Universiteit Delft, 2015.

L. Lamorgese and C. Mannino. An Exact Decomposition Approach for the Real-Time Train Dispatching Problem. *Operations Research*, 63(1):48–64, 2015.

L. Lamorgese and C. Mannino. A non-compact formulation for job-shop scheduling problems in traffic management. *Operations Research*, pages 1–38, 2019.

L. Lamorgese, C. Mannino, and M. Piacentini. Optimal Train Dispatching by Benders'-Like Reformulation. *Transportation Science*, 50(3):910–925, 2016.

A. Mascis and D. Pacciarelli. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143(3):498–517, 2002.

J. Naoum-Sawaya and S. Elhedhli. An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, 210(1):33–55, 2013.

S. Narayanaswami and N. Rangaraj. Scheduling and Rescheduling of Railway Operations: A Review and Expository Analysis. *Technology Operation Management*, 2(2):102–122, 2011.

M. Peeters and L. Kroon. Circulation of railway rolling stock: a branch-and-price approach. *Computers and Operations Research*, 35(2):538–556, 2008.

J. Rodriguez. A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B: Methodological*, 41(2):231–245, 2007.

A. Sarac, R. Batta, and C. M. Rump. A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research*, 175(3):1850–1869, 2006.

A. Schöbel. Capacity constraints in delay management. *Public Transport*, 1(2):135–154, 2009.

S. Sen and H. D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223, 2006.

H. D. Sherali and B. M. P. Fraticelli. A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22:319–342, 2002.

A. Toletti. *Automated Railway Traffic Rescheduling and Customer Information*. Phd thesis, ETH Zurich, 2018.

J. Törnquist and J. A. Persson. Train Traffic Deviation Handling Using Tabu Search and Simulated Annealing. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 00(C):1–10, 2005.

B. Tsinghua and C. Yan. Mixed-integer Programming Based Approaches for the Movement Planner Problem : Model , Heuristics and Decomposition. In *RAS Problem Solving Competition*, pages 1–14, 2012.

T. J. J. Van den Boom and B. De Schutter. Modelling and control of discrete event systems using switching max-plus-linear systems. *Control Engineering Practice*, 14(10):1199–1211, 2006.

F. Vanderbeck. On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48(1):111–128, 2000.

F. Vanderbeck. Implementing mixed integer column generation. In *Column Generation*, pages 331–358. Springer Boston, MA, 2005.

F. Vanderbeck and M. W. Savelsbergh. A generic view of Dantzig-Wolfe decomposition in mixed integer programming. *Operations Research Letters*, 34(3):296–306, 2006.

F. Vanderbeck and L. A. Wolsey. An exact algorithm for IP column generation. *Operations Research Letters*, 19(4):151–159, 1996.

L. A. Wolsey. A Rescource Decomposition Algorithm for General Mathematical Programs. *Mathematical Programming*, pages 244–257, 1981.

L. A. Wolsey and F. Vanderbeck. Reformulation and decomposition of integer programs. In *50 Years of Integer Programming 1958-2008*, pages 431–502. Springer, 2010.

X. Zhou and M. Zhong. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B: Methodological*, 41(3):320–341, 2007.