A Set Packing Inspired Method for Real-Time Junction Train Routing

Lusby, Richard Martin; Larsen, Jesper; Ehrgott, Matthias; Ryan, David

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Lusby, R. M., Larsen, J., Ehrgott, M., & Ryan, D. (2009). *A Set Packing Inspired Method for Real-Time Junction Train Routing*. DTU Management. DTU Management 2009 No. 15
http://www.man.dtu.dk/upload/institutter/ipl/publ/publikationer%202009/rapport%2015.pdf

A Set Packing Inspired Method for Real-Time Junction Train Routing

Report 15.2009

DTU Management Engineering

Richard M. Lusby
Jesper Larsen
Matthias Ehrgott
David Ryan

December 2009

DTU Management Engineering
Department of Management Engineering

A Set Packing Inspired Method for Real-Time Junction Train Routing

Richard M. Lusby[‡][*], Jesper Larsen[‡], Matthias Ehrgott [§], David Ryan [§]

[‡] Department of Management Engineering, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
[§] Department of Engineering Science, The University of Auckland, Auckland, New Zealand

Abstract

Efficiently coordinating the often large number of interdependent, timetabled train movements on a railway junction, while satisfying a number of operational requirements, is one of the most important problems faced by a railway company. The most critical variant of the problem arises on a daily basis at major railway junctions where disruptions to rail traffic make the planned schedule/routing infeasible and rolling stock planners are forced to reschedule/re-route trains in order to recover feasibility. The dynamic nature of the problem means that good solutions must be obtained quickly. In this paper we describe a set packing inspired formulation of this problem and develop a branch-and-price based solution approach. A real life test instance arising in Germany and supplied by the major German railway company, Deutsche Bahn, indicates the efficiency of the proposed approach by confirming that practical problems can be solved to within a few percent of optimality in reasonable time.

Keywords: Train Routing, Disruption Management, Duality, Optimization

1 Introduction

Disruption Management is one of the most important fields within Operations Research. Broadly speaking, this area of research deals with situations in which a predetermined "optimal" operational plan is exposed to some form of disruption making the plan sub-optimal, perhaps even infeasible. To manage the disruption effectively, one must re-optimize the original plan while minimizing the negative impact of the disruption. Efficient disruption management tools are essential to the success of any operation and have been employed in a wide range of industries. For example, [5, 6] and [17] describe disruption management in the airline industry, while [28] details applications of disruption management approaches in production planning and the telecommunications industry.

In this paper we consider an important operational problem from the railway industry. This problem, one variant of the more general problem of routing trains through railway junctions, arises at complex junctions of a railway network and requires one to re-route/re-schedule the set of timetabled trains when a disruption occurs. Typical examples of disruptions include late train arrival, track maintenance, or even accidents, all of which will impact the predetermined operating plan for the junction by enforcing route changes and/or additional delays to trains. Efficiently coordinating train movements on a junction in a disruption recovery setting is of crucial importance

[*]Corresponding author. E-mail address: rmlu@man.dtu.dk

to railway companies as it is a regularly occurring problem and any delays incurred by rail traffic will undoubtedly propagate through the timetable due to the interdependent nature of train movements. For many countries, an extensive rail network forms the backbone of the transport system, and this can be quickly disrupted if delays are mismanaged. Furthermore, the efficiency of a rail service is quite obviously positively correlated with the number of trains that arrive and depart on time. The data used in this paper was provided by the German railway company, Deutsche Bahn. Deutsche Bahn is a state-owned company and the largest provider of rail services in Germany. Employing around 230,000 people, it operates just over 28,000 passenger trains and about 4,700 freight trains daily on a rail network that totals some 34,000 kilometres in length and connects approximately 5,700 stations. In 2005, Deutsche-Bahn carried approximately 120 million passengers on its long distance train services (at an average distance of 280 kilometres) and approximately 1.7 billion passengers on its short distance train services (at an average distance of 20 kilometres). In the same year, some 250 million tonnes of freight were transported at an average distance of 310 kilometres (see [11]). Given the sheer size of its rolling stock routing problems, it is essential that Deutsche Bahn has effective tools to minimize the impact of disruptions when they occur.

The problem of routing trains through railway junctions is an important planning problem faced by a railway company and arises at each of the strategic, tactical, and operational planning levels. While the strategic and tactical level variants, which address junction capacity and timetable feasibility, respectively, have been well studied (for the different approaches see e.g. [4, 12, 19, 23, 30]), the operational variant has received relatively limited attention in the literature. In this paper we present a set packing inspired formulation of this problem and describe a branch-and-price based solution approach. Our formulation is characterized by a resource based constraint system that allows one to explicitly represent the movement of trains over time on a junction. The objective of the proposed model is to recover feasibility of a disrupted routing while minimizing the propagation of delays. Here, as in many real-time problems, feasibility is emphasized more than optimality. We highlight the flexibility of the model by showing that it can easily incorporate both spatial and/or temporal changes to train movements. The complete approach is tested on 180 real life instances arising from a medium sized junction of the German rail network.

This paper is structured as follows. Section 2 introduces the problem considered in more detail and gives an overview of the operational requirements of the German railway network. Section 3 provides a review of relevant literature and this is then followed in Section 4 by a detailed description of our integer programming formulation of the problem. Section 5 focuses on the proposed branch-and-price based solution approach which utilizes ideas from duality theory and column generation, while Section 6 summarizes our computational results. The main conclusions of this study as well as directions for future research are given in Section 7.

2 Problem Definition

To establish the necessary context for the problem considered in this paper, we begin by giving a brief description of railway networks in general as well as the particular operational requirements of the German rail network. For a more detailed explanation of the latter the reader is referred to [19] and [22]. A railway network in its simplest form can be viewed as a graph consisting of several nodes and edges. Each node corresponds to a component of the rail network where significant train interaction occurs and is itself a complex network of track that allows trains to change direction, stop, or possibly overtake other trains. The nodes of the network are what we refer to as junctions. Often a junction coincides with the location of a station; however, station-less junctions also exist (see [12] for an example). A station will always have one or more platforms where passengers can board and disembark trains. An edge between any two nodes indicates the section of track interlinking them. This typically consists of parallel stretches of one-way track, where each direction is dedicated to rail traffic in a particular direction. The terminology as it applies to the German railway network is given in Figure 1.

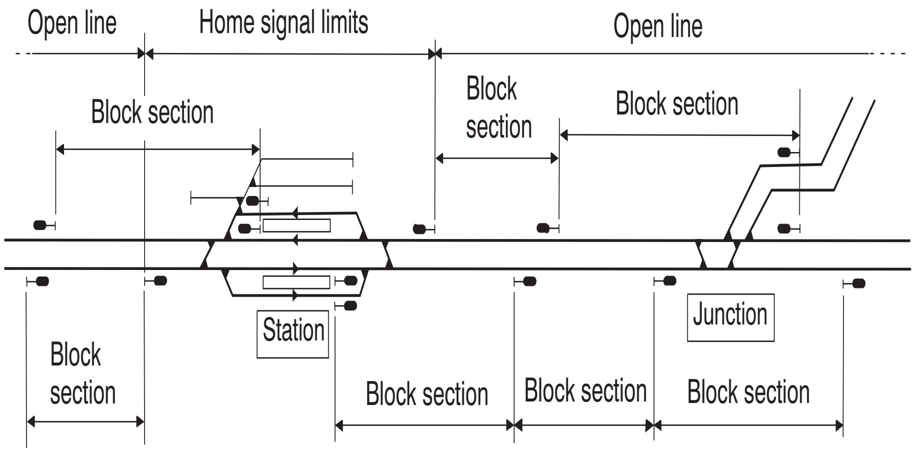

Figure 1: An example of the German railway network (taken from [22])

In Germany, railway track is classified as either *station track* or *tracks of the open line*. Station tracks are those defined to be within the boundaries of a railway station and will always include platforms. The boundary of a railway station is clearly indicated by *home signals*. Tracks of the open line, on the other hand, connect stations and may include junctions.

Train movements on a railway network are typically controlled by a signalling system. Such a system prevents trains from getting too close to one another by ensuring that a minimum amount of time (known as *headway*) separates any pair of trains that have a section of track in common. To implement a signalling system, tracks of the open line are divided into so called *block sections* on which there can be at most one train at any given time. The entrance to block sections is controlled by block signals that indicate whether the subsequent block section is occupied or vacant. Note that the station track leading from a home signal to a platform (depicted as rectangles within the station in Figure 1) is not referred to as a block section but rather an *interlocking section*. It is, however, used in much the same way. Both block and interlocking sections may contain a number of smaller *track sections*. This is due to the fact that block (interlocking) sections may contain track that is contained in other block (interlocking) sections. For instance they may share a *switch* or an *intersection*. A switch is a track device that allows a train to change railway lines, while an intersection permits two sets of tracks to cross one another. Track sections denote the boundary of infrastructure that is common to multiple blocks. When a train enters a block (interlocking) section, the entrance time of all track sections within the block is synchronized with that of the first (i.e. the entrance times of all track sections within the block is equal to that of the first), and each is successively released once the tail of the train has exited (and some additional buffer time has elapsed). Released track sections may then be claimed by other trains. Two train paths are said to be in *conflict* if they simultaneously claim the same track section. The term “claim” is used as, due to the signalling system, it is possible for two trains to compete for the same track section simultaneously even though they may not necessarily occupy the track section at the same time. Note that track sections may overlap one another, meaning that the same physical piece of track can be defined by multiple track sections; this is particularly true in very congested junctions. For this reason a train is required to claim additional track (although it will not actually traverse it) in order to prevent conflicting movements onto the tracks the train will traverse. Track sections that are claimed but not actually traversed a termed *banned* track sections.

It is at the nodes of a railway network where the problem considered in this paper arises. Depending on the number of switches on a junction, a train is likely to have several possible *routes* it may use in traveling from its so called *entering point* of the junction to its designated *leaving point*. A route is defined to be a sequence of track sections, while the entering and leaving points collectively delimit the infrastructure considered. This is typically just the junction; however, it may also include some part of the surrounding network. Furthermore, we introduce variability in the time it takes a train to traverse a given route by considering different acceleration and deceleration strategies for trains that can traverse the route. We define the term *path* which is train type dependent and refers to the traversal of a given route over time. The fundamental goal of any junction train routing problem is to identify the best set of conflict-free paths on the junction infrastructure for a timetabled set of trains (i.e. known arrival and departure times) while satisfying the operational requirements above and possibly several other service constraints (i.e. coupling/decoupling of trains). For the particular problem in this paper, the best set of paths is considered to be the set of paths that minimizes the total deviation, possibly weighted by train priority, from the arrival and departure times of the trains given a disruption.

3 Literature Review

In this section we review contributions in the area of real-time train scheduling and routing. Without providing an exhaustive review, we attempt to provide an overview of the models and methods that have been adopted in the literature. Cordeau et al., in [7], survey a large number of papers in this field and a more recent paper by [26] reviews several important contributions since then. The studies cited in both papers tend to focus on single track networks, or more complicated networks in which the routing of the rail traffic through the station/junction is ignored. The difficult junction train routing problem has only appeared in the last decade or so and, moreover, has really only been applied at the strategic and tactical planning levels (see [20] for a comprehensive survey). Here we describe some of the most common approaches as well as focus on those applications that consider more complicated routing situations.

Higgins et al. present a mixed integer programming (MIP) model in [15] for the scheduling of trains on single track networks. Such networks are commonly found in the freight industry, cater for bidirectional train movements, and have several so called *sidings* to facilitate train passing. Binary variables are used to dictate the ordering of trains on the sections of track between two sidings, while continuous variables are used to model train arrival and departure times at sidings. The authors permit variability in track traversal time and the constraint set enforces the temporal restrictions trains must respect. The model is solved using branch-and-bound with an objective that minimizes total train delay weighted by train priority. Instances with up to 30 trains and 12 sidings are reported. In a subsequent paper, [16], Higgins et al. develop and compare several heuristic techniques. Instances having 50 trains with between 103 and 113 conflicts are considered.

The same problem is considered in [3] where the authors present a two-phase heuristic to solve the problem. In the first phase, the current time and position of all trains is updated so that each is positioned at a siding. The second phase implements a refined version of the greedy heuristic in [2], which considers trains in chronological order and resolves conflicts at the local level only. The heuristic attempts to minimize the total delay in scheduling the trains. An implementation of the algorithm is reported for an Asian railway company, where up to 400 trains run per day with as many as 60 in the system at any given time.

In [8], Şahin also considers the real-time scheduling of trains on a single track network and presents a heuristic that considers conflicts in chronological order and sequentially resolves them. For both trains involved in a conflict the algorithm considers the delay necessary to resolve the conflict. This is combined with a look ahead feature that determines the expected arrival of all other trains at their respective destinations based on each delay option. The train that gives the least expected delay to the rest of the trains is delayed. Computational experiments for between 6 and 20 trains on a 163 kilometre stretch of track in Turkey are given.

A similar problem is the focus of [1]. The MIP model presented is considered to be too complicated to solve in real-time, and the authors adopt a more practical heuristic approach. A disruption that creates train conflicts is assumed to affect a number of *services*, where a service corresponds to the movement of a train between two stations. The heuristic aims to reassign trains to services so that the total delay is minimized and is built on a tree enumeration approach.

Oliveira and Smith, in [21], propose a job shop scheduling formulation for managing disruptions on a single track railway. The model is solved using constraint programming techniques. Train conflicts are considered and resolved in chronological order with the objective of minimizing total delay. The algorithm includes a number of real world constraints. In particular, it is possible to force two trains to reside at the same station for some overlapping time interval and to allow the same train to perform multiple itineraries.

A greedy travel advance strategy based on a discrete event model is described in [13] for the scheduling of trains on a single line. The proposed algorithm includes a nonlocal capacity check to avoid deadlocks and can efficiently handle time perturbations to the schedule. The authors show that the approach easily extends to networks with double track sections, can handle heterogeneous rail traffic, and performs well on three time-performance criteria. A fictitious network containing seven nodes and 36 trains over a 12 hour period is considered.

Törnquist and Persson, in [27], consider the re-scheduling of trains under disturbances on so called N-tracked networks. This extends the single track network by allowing certain sections of track to consist of multiple parallel sections, each of which can accommodate one train. The authors present a MIP formulation of the problem and describe four different solution approaches. A real-life application from the Swedish rail network is used to test the proposed methodology. The network tested contained 169 interlinked stations while the daily timetable considered had 92 freight and 466 passenger trains. Computational experiments assume a single delayed train.

With the exception of [9], [10], [23], and [24], there appear to be very few studies that consider the real-time scheduling/routing of trains on more complicated networks, such as junctions. In [23] and [24], the authors consider a complicated station-less junction near Paris in France. A job shop scheduling formulation is proposed and solved using constraint programming techniques. The model described is an extension of that proposed in [12], for the capacity assessment of rail traffic on junctions. In particular, a variable waiting time is introduced on the traversal time of track sections to permit trains to wait at certain points on the junction. Given the nature of the formulation, re-routing options are almost never considered. This approach attempts to find the minimum total delay necessary in keeping the trains on their assigned paths. Instances with between 6 and 24 trains are considered.

Job shop scheduling inspired approaches are also described in [9] and [10]. Both papers model train paths using alternative graph methodology. A pair of alternative arcs is used to enforce a train sequencing order at any block section where two trains are in conflict. An alternative graph is built using one path per train. The authors of [10] show that the longest path in this graph is equivalent to the maximum propagated delay of the corresponding train sequencing. A truncated branch-and-bound algorithm is described to minimize the length of this path. The method is extended in [9] to include a local re-routing strategy. The iterative procedure described first computes an optimal sequencing of trains for a given set of routes and then, using the local re-routing strategy, attempts to improve this solution by re-routing some trains. Practical problems from a section of the Dutch rail network with up to 40 trains per hour are considered.

A complementary notion to that of disruption management is *robustness*. Generally speaking, the concept of robustness refers to how susceptible an operational plan is to disruption. Viewed in the junction train routing context, a robust routing is one for which delays to trains are less likely to propagate. By considering robustness at the tactical level one can minimize the dependence on disruption management tools. We do not go into detail here, but instead refer the reader to [4],[12], and [14] for ways of including robustness when determining train routes at the tactical level.

Whether it be single track networks or more complicated junctions, the most common approach when allocating paths to trains is to first identify those track sections where conflicts occur and then explicitly impose restrictions on the sequencing of trains to prevent the conflict. Examples of this include the 0-1 binary decision variables which are usually found in MIP formulations for the single track case and govern the ordering of pairs of trains on track sections, the disjunctive constraints of the job shop scheduling formulations in [23] and [24] that ensure trains do not use the same track section at the same time, and the alternative arcs described in [9] and [10]. An approach that implicitly resolves track based conflicts is absent in the literature. That is, a formulation of the problem which does not require the a priori identification of conflicts. The resource based set packing inspired formulation of Section 4 is the first approach that achieves this.

4 Model

In this section we formulate the junction train routing problem as a mixed integer program with a resource (i.e. track section) based constraint system. As will become clear, such an approach does not require the explicit identification of all potential train conflicts and is also very flexible in that it can easily accommodate both spatial and temporal changes to train movements. Recall that to provide a conflict free routing through a junction for a set of trains, one must ensure that at most one train is claiming any track section of the junction at any given time. Given both the spatial and temporal requirements in such a restriction, any resource based constraint system must naturally consist of a component in each dimension. The most logical approach is to suitably discretize the time horizon one is considering into a set of much shorter time intervals and define a constraint for each track section in each time interval. That is, the set of track sections is multiplied by the number of time intervals and, by doing so, one effectively observes the entire junction at recurring time intervals. We use the term *tints* for a time interval track section pair. This approach of monitoring a junction over time is consistent with what is done in practice. The duration of the time interval is, however, likely to be railway company dependent and here we use the value of 15 seconds. Each tints resource is assigned a capacity of one.

Given the resource based constraint system defined above, it is easy to describe a train path. Any train path can be represented as a column with the following components. A one in a particular row indicates that the train is claiming (and will definitely traverse) the track section at the time specified by the tints resource, while a zero indicates that the train is not claiming the tints resource. The inclusion of banned track sections complicates the modelling as it is possible for two trains to ban the same tints resource yet not be in conflict. Consider the example given in Figure 2. Both train movements (given by the arrows) are obviously not in conflict; however, each bans the track section connecting the two lines to prevent conflicting movements onto each of the lines.

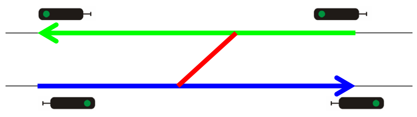
Figure 2: A banned track section

As a consequence, one cannot specify that banned track sections be covered with value one;

this is too restrictive and may prevent feasible train paths from being selected. We therefore specify that banned track sections are covered at value $\frac{1}{2}$. The value of $\frac{1}{2}$ is chosen as it is extremely unlikely that three or more trains will simultaneously ban the same track section.

With the general structure of the model outlined above, one can formally define the MIP model as follows. Let us assume we have a set of trains N ($|N| = t$), that each train $i \in N$ has n_i possible paths through the junction (including the so called *null* path which corresponds to the train not being routed), and that $\sum_i^t n_i = n$. Let us further assume we have a set of tints resources S ($|S| = l$). We denote the set of all columns by Ω ($|\Omega| = n$) and define the two matrices T and R, each of which consists of n columns. Matrix $T = (T_{i\omega})$ contains a row for each train, and $T_{i\omega} = 1$ if and only if column ω defines a path for train $i \in N$. Matrix $R = (R_{s\omega})$ contains a row for each tints resource $s \in S$. As explained above, element $R_{s\omega}$ can take one of the three following values: 1 if path ω uses tints resource $s \in S$, $\frac{1}{2}$ if path ω only bans tints resource $s \in S$, and 0 otherwise. The cost, c_ω, of any path $\omega \in \Omega$ for train $i \in T$ is the deviation, δ_ω, of the path (in seconds) from the train i's scheduled arrival time weighted by train i's priority, ρ_i. The value of δ_ω is calculated as $\max(0, \alpha_\omega - a_i)$, where a_i is the scheduled arrival time of train i. The scheduled arrival time usually pertains to the train's arrival time at the platform; however, it could also be the train's earliest junction exit time if the train has no scheduled stop. The actual arrival time of the train at such points on path ω is given by α_ω. On the occasion that the train has more than one scheduled stop when travelling through the junction, the deviation of the path will be the train's deviation at its last scheduled stop. The inclusion of the priority parameter is to reflect the importance of each train in the problem. Finally, the binary decision variable x_ω is equal to one if path ω is used in the solution and is zero otherwise. The full formulation is given below.

$$\text{Minimize:} \qquad \boldsymbol{c}^T \boldsymbol{x} \tag{1}$$

Subject To:

$$T\boldsymbol{x} = \mathbb{1} \tag{2}$$

$$R\boldsymbol{x} \leq \mathbb{1} \tag{3}$$

$$\boldsymbol{x} \in \{0,1\}^n \tag{4}$$

The objective function minimizes the total deviation weighted by train priority for all trains. Constraints (2) ensure that all trains receive a path, while constraints (3) enforce the restriction that at most one train can claim any track section during each time interval. Finally, constraints (4) give the binary restrictions on each of the decision variables. One can observe that the inclusion of the $\frac{1}{2}$ elements in R destroy what would otherwise be a set packing model. One can further observe that the addition of constraints (2), so-called GUB constraints, guarantees that the extreme points of the polytope defined by the constraint system and the submatrix consisting of columns for a single train (train block partition) are integer solutions. It is trivial to see that the addition of such a constraint dominates all others in the train block partition. This ensures that fractions will only occur as a result of two different trains competing for the same tints resource.

Given the nature of the constraint system, it is obvious that the model implicitly obtains a conflict free set of paths for the trains. Furthermore, one can also dynamically consider additional paths for trains without any fundamental change to the model itself. Additional paths can easily be included as extra columns in the model. We exploit this property when we develop an efficient solution approach in Section 5. In what follows we describe how to construct the variables of this model (Section 4.1), analyze the structure of a basic feasible solution (Section 4.2), and finally introduce the concept of the dual representation of a basic feasible solution (Section 4.3).

4.1 Variable Generation

The columns given in the model (1)-(4) define the movement of trains through topological points of a railway junction over time. Each variable thus states, to the nearest time interval, the claim and release time of each track section of its underlying route. Any train path is hence a simulated movement for a train based on certain acceleration and deceleration strategies. The variable generation phase is governed by the following three important factors: the junction infrastructure, the kinematic capabilities of the trains, and the signalling system in place at the junction. All three naturally impose restrictions on the movement of trains. The junction infrastructure dictates the speed limit that trains must adhere to, the kinematic capability of a train specifies how fast it can accelerate or decelerate (among other things), and the signalling system specifies how far in advance a track section must be claimed (banned). The generation of a set of train paths entails enumerating all possible ways the train may traverse the underlying route taking into consideration all the above factors. This is achieved by constructing a tailored time-space network

When initializing the path generation process for a particular route we assume that both the train's arrival time and arrival speed are known and that the train is not accelerating or decelerating. The initial speed is assumed to be the train's maximum speed or the maximum permitted speed on the first track section of the underlying route, whichever is the smaller. This initial condition defines the starting point of all feasible paths on the route and can be represented as a single time-space label. We further assume that the driver of the train will only consider altering the speed of the train on entrance to specific track sections. In particular, only those track sections that mark the beginning of a new block section or a new speed limit. The speed of a train is something that is manually controlled, and the driver will need some form of visual landmark that provides information on when the speed can be changed. Moreover, when altering the train's speed the driver is restricted to the following: 1) proceeding with constant velocity, 2) accelerating at a constant rate, or 3) decelerating at a constant rate. We acknowledge that constraint rates of acceleration and deceleration is slightly unrealistic; however this is consistent with what is done in practice (see [18],[23], and [29]).

Given the initial label, subsequent labels are generated based on the kinematic options above using well known kinematic formulae and each represents the train entering a particular track section at a certain time. Prior to extending the first label, a preprocessing step adjusts the speed limit on each of the track sections to ensure trains to not reach speeds that make them unable to comply with the speed limit of subsequent track sections. When generating a deceleration label, if the train can come to a complete halt on the track section it is entering, the deceleration is performed in such a way that the train comes to a stop on entry to the next track section. A terminating condition for label generation is specified in the form of a maximum path duration. This is set prior to running the path generation, and only paths with a duration shorter than this are generated. Note that labels which correspond to a train waiting on a track section are not defined. That is, if a train reaches a speed of zero, it immediately accelerates. Waiting on track sections can be achieved by pushing a time shift through components of such a time-space network, and this is discussed in Section 5. The only restriction is that the path duration limit is sufficiently long to at least allow the generation of labels which correspond to the train stopping on entrance to the block sections of the route. If it is a requirement for the path that the train stops at one (or more) platforms, then this can be easily included. To model a train stopping at a platform, one requires the entrance speed of the train on arrival at the track section immediately following the platform track section to be zero. Only labels that satisfy this condition are extended. The entrance time of such labels must also be modified to include the train's dwell time at the platform.

To control the number of labels generated, two different aggregation strategies are implemented. The first one considers label aggregation. For two labels that correspond to the train entering the same track section during the same time interval, if the respective entrance speeds are within $1ms^{-1}$, then only one of the two labels is retained; labels that satisfy the above conditions are considered to be sufficiently similar. Since a track section by track section approach is adopted for label extension (i.e all labels for one track section are extended before its subsequent track section), this aggregation strategy can be routinely performed at track sections marked as merging points. Merging points are predetermined and indicate track sections where time-space label merging will occur. For example, every track section on the route could be a merging point. The second strategy, on the other hand, aggregates track sections. As was mentioned in Section 2, the boundary of a track section often acts as a release point and simply indicates that a component of a claimed block section may be released. At such points it is unlikely that the train will consider changing speed. Hence all consecutive track sections delimited by a release point are aggregated and considered as one longer track section. Other restrictions on label generation are also incorporated to prevent the train accelerating once it has commenced deceleration and vice versa. Associated with each label is information on where (i.e. which track section) the rear of the train is on. For consecutive labels, this information can be used to determine which track sections have been released in traversing the track section(s) defined by the former label. Given the nature of the network, it is very difficult to construct identical labels via completely different traversal strategies and, for this reason, the resulting time-space network is a tree structure. Algorithm 1 summarizes the steps involved in constructing the time-space network for any route.

Algorithm 1: Route Time-Space Network Generation

```
1  Initialization: Define the initial label and add to label list;
2  Preprocess track section speeds;
3  while label list is not empty do
4      Get the next label to extend;
5      if Label track section is a merging point and not already merged then
6          Merge labels in list;
7      if Track section aggregation is possible then
8          Aggregate track sections for label;
9      if Acceleration is possible and extension is feasible then
10         Generate label for acceleration and add to label list;
11     if Constant velocity is possible and extension is feasible then
12         Generate label for constant velocity and add to label list;
13     if Deceleration is possible and extension is feasible then
14         Generate label for deceleration;
15         if Track section is a designated platform stop then
16             if Entrance speed of extension label equals zero then
17                 Add dwell time to entrance time of extension label;
18                 Add modified extension to label list;
19         else
20             if Entrance speed of extension label equals zero then
21                 if Track section marks entry to block section then
22                     Add extension to label list;
23             else
24                 Add extension to label list;
```

All possible paths on a given route are contained in the corresponding time-space network and can be constructed from the respective leaf label by tracing one's steps back through the tree.

4.2 Structure of a Basic Feasible Solution

In this section we consider the structure of any basic feasible solution to the LP relaxation of the model (1)-(4). We identify certain properties that dictate what solution method will be most effective. We denote the total number of constraints of the model as m, where $m = t + l$. While t can be expected to be relatively small, one can expect l to be quite large. The latter is a function of both the number of time intervals and the number of track sections. An increase in either of these parameters will create a large number of additional tints resources. For any basic feasible solution, the $m \times m$ basis matrix will include $\tau \geq t$ train path variables. The remaining $(m - \tau)$ variables will correspond to slack variables on the tints resource constraints. Hence, any basis matrix B is of the form

$$B = \begin{bmatrix} T_B & 0 \\ P_B & S_B \end{bmatrix},$$

where T_B is a $(t \times \tau)$ 0-1 matrix corresponding to the GUB constraints, P_B is an $(l \times \tau)$ matrix defining the train paths, and S_B corresponds to the $(m - \tau)$ slack variables. The matrices T_B and P_B collectively define the the *train path partition* of B. The presence of slack variables in the basis, each with a value in the interval $(0, 1]$, indicates that the corresponding tints resource constraints are non-binding and thus have dual variable values of zero. At most (τ) dual variables can take non-zero values, and $(\tau - t)$ of these will correspond to binding tints resource constraints where the τ basic train path variables are fully utilizing or even competing for the use of the tints resource. Since the number of such binding constraints can be expected to be small for a set of τ basic train paths, we can expect that τ is unlikely to exceed a small multiple of t. With such a large basis matrix, the conventional revised simplex approach of using B^{-1} to calculate the dual variables at each iteration is expected to be computationally expensive. Alternatively, one could filter the basis matrix and only retain those tints resource constraints where conflicts occur, but this is likely to be impractical if there are too many variables to consider or if the model needs to be implemented in a dynamic environment. In Section 4.3 we introduce the concept of the dual representation of a basic feasible solution which circumvents the computational expense of using a large basis yet retains the flexibility of being dynamically modified.

4.3 Dual Representation of a Basic Feasible Solution

The dual variables for any basic feasible solution can also be calculated as the solution of the following LP known as the dual representation of the train path partition:

$$\text{Maximize:} \qquad \boldsymbol{\pi}^T \mathbb{1} + \boldsymbol{\mu}^T \mathbb{1} \tag{5}$$

Subject To:

$$T_B^T \boldsymbol{\pi} + P_B^T \boldsymbol{\mu} \geq \boldsymbol{c_B} \tag{6}$$

$$\boldsymbol{\mu} \geq 0, \tag{7}$$

where $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ are the dual variables on the GUB and tints resource constraints, respectively, and $\boldsymbol{c_B}$ is a vector containing the costs of the basic train path variables. A basis for this LP has dimension $(\tau \times \tau)$, which can be expected to be very much smaller than the $(m \times m)$ basis matrix B. Note that in forming the LP (5)-(7) one only requires a subset of all train path variables. Applying simple duality theory one can observe that updating basis B using the standard revised simplex approach through the addition of an entering train path variable is equivalent to adding a violated constraint to the dual representation of the train path partition of B and reoptimizing, the magnitude of violation being equivalent to the reduced cost of the entering variable. This revised simplex modification forms the core of our solution approach and will be discussed in more detail in Section 5, where we develop an efficient method for solving model (1)-(4).

5 Solution Methodology

When one solves the disruption recovery version of the junction train routing problem, one has, as input, a conflict-free set of train paths. This solution has been obtained at the tactical level and will, providing no disruption occurs, be feasible throughout the execution of the daily operations. Given one or more disruptions, one must adjust the paths of the affected trains such that the impact of the disruption is minimized. Here we exploit the potential for model (1)-(4) to be dynamically updated by first considering only a small number of train paths and then gradually expanding this model through the inclusion of additional train paths while infeasibility persists. Infeasibility is reflected by the presence of one or more null paths in an optimal solution to the LP relaxation of model (1)-(4). In what follows we describe several components of the methodology. In particular, we discuss: how the the optimal solution to the LP relaxation of model (1)-(4) is obtained using model (5)-(7), how additional routes and delays are introduced for trains, and finally how integerality of the solution is obtained.

5.1 Solving the LP Relaxation

As discussed in Sections 4.2 and 4.3, we prefer to calculate the dual variables for a basis of the form B by solving the much smaller LP formed from the dual representation of the corresponding train path partition. This approach has the advantage that very few train path variables are required to be explicitly in the model (only those defining the basis as well as a subset of non-basic variables). All other variables are implicitly stored “off-line” as tree structures and are routinely “priced” to identify favourable train paths.

This decomposition approach appears noticeably similar to column generation in that we have identified both a kind of master problem and a pricing problem; however, unlike traditional column generation, the pricing problem is not formulated as an optimization problem. Rather, an enumerated set of train paths currently under consideration is contained in the relevant tree structures (see Algorithm 1) and an efficient, recursive tree traversal algorithm can be used to price the paths on any route. Since each label of a route tree structure contains information on what track sections have been released since its predecessor label, by storing an accumulated dual variable value at each label one can easily identify leaf labels (and hence paths) with a negative reduced cost. That is, by pricing the tree structures with vectors $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$, one can identify entering train path variables. Questions arise as to how many negative reduced cost train paths one should return at any pricing iteration. Here we implement the multiple pricing approach of returning the best path on each route for every train. This approach is shown to have a positive impact on the convergence of the LP relaxation for model (1)-(4) in [19]. Furthermore, the dual simplex algorithm is used to re-optimize the dual representation of the train path partition on appending any entering train path variables as violated constraints. The main ideas of the solution approach are summarized in Figure 3.

Note that the initialization step requires one to construct an initial set of tree structures as well as identify the initial train path partition. The possibility of dynamically increasing the number of tree structures in the pricing phase is indicated in Figure 3. For the initial train path partition, one can use the trivial solution where each train is assigned its null path; however, to provide more accurate dual information for the pricing routine, we implement a greedy construction heuristic that considers the trains in chronological order of arrival times and attempts to assign each train

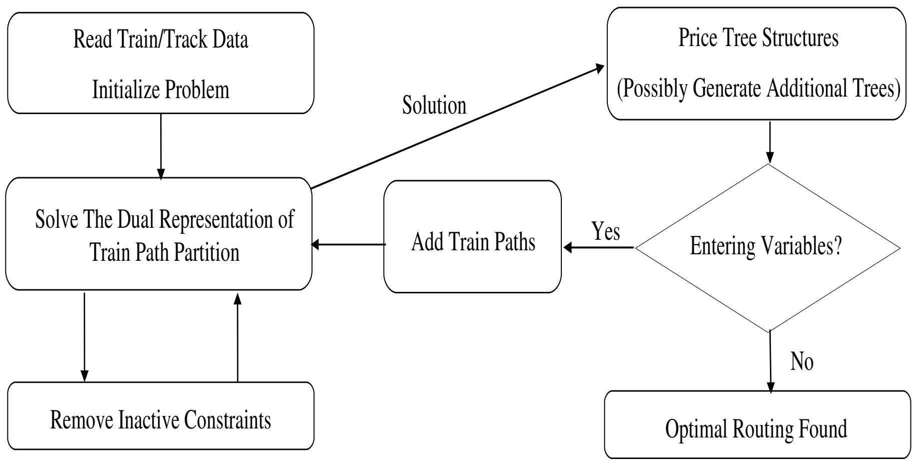

Figure 3: Solving the LP

its best path. If the train's best path is not available, the train receives its null path. Note that one also has the possibility of removing inactive constraints from any optimal solution to the dual representation of a train path partition in order to keep the row dimension of this problem small. Inactive constraints indicate that the corresponding train path variable is non-basic and can be removed. Computational experiments in [19] indicate that this tends to have a detrimental effect on the LP relaxation convergence time of model (1)-(4) as train path variables tend to reappear once removed. For this reason, we do not consider removing inactive constraints of model (5)-(7) during the LP convergence of model (1)-(4). This is, however, required when forcing integrality using branch and bound.

5.2 Additional Train Path Variable Construction

As has been previously stated, the solution to the junction train routing problem found at the tactical level will be feasible providing there are no disruptions. If, under disruption, this planned schedule is infeasible, one must provide the affected trains with extra flexibility in an attempt to restore feasibility and minimize the impact of the disruption. Temporal flexibility can be provided in the form of delayed train paths (i.e. allowing trains to wait at certain points on their routes) or spatially in the form of additional routes. Each possibility is discussed in turn and we conclude with a discussion on how, based on a disruption, the problem is expanded.

5.2.1 Generating Delayed Train Paths

Given the nature of the route tree structures, one can observe that the set of paths for a particular route remains valid, in terms of the kinematic capabilities of the train, regardless of the junction entrance time of the train. That is, irrespective of what time the train enters the junction, the acceleration and deceleration strategies available to the train are unchanged. The only difference appears in the time dimension of the network, where all times are shifted forward or back depending on the train's deviation from its scheduled entrance time. This approach of pushing a time shift through the a tree structure can also be used to model a train waiting at a block section on a junction. Here, however, the time shift will only occur in a component of the relevant tree structures and be equal to the waiting time of the train.

To permit delayed train paths to be considered, whether it be on entrance to the junction or at main signals on the junction, one can modify the tree traversal pricing routine described in Section 5.1. Pricing a tree structure involves a recursive tree traversal that assigns an accumulated dual variable value at each label. When waiting at block signals on the junction is not permitted, each leaf label corresponds to a unique path. This remains unchanged when we permit trains to only be delayed on entrance to the junction as all claim and release times of the track sections on any path will simply be translated forward in time by the magnitude of the entrance delay. If, however, one also permits trains to wait on block sections on the junction, this is no longer true. In a disruption recovery situation, we permit trains to wait for increments of 150 seconds, termed a *halting interval*, at a maximum of three block sections on any path. The three permitted stops also includes any scheduled stops the train has as we always prefer to increase the dwell time at a platform as opposed to making the train stop at some intermediate point on the junction. Hence, depending on where, and for how long a train waits, each leaf label may correspond to multiple paths. The acceleration and deceleration strategies will be the same; however, the accumulated delay, or the main signals where the train is forced to wait will definitely be different. The modified

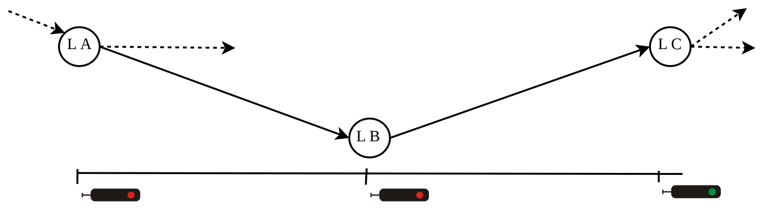

Figure 4: Modelling a delayed train movement on a junction

tree structure pricing routine is further explained by Figure 4. This illustrates a component of a tree structure corresponding to the train stopping on entry to a particular block section (given by label LB). It is precisely here where one has the ability to allow the train to wait for a number of halting intervals and then push this time shift through the subtree originating from label LB. To permit a train to wait for a various number of halting intervals at a main signal, one must implement a nested tree traversal. Each leaf label then corresponds to more than one possible path. Note that this nested tree traversal routing does not remove the possibility of not delaying trains.

5.2.2 Additional Route Inclusion

One also has the possibility of providing trains with additional routes. In a disruption recovery situation, each train will have a number of possible routes that it can use to avoid any conflicts and travelling through the junction. Some will be more preferable than others. In particular, routes that have less switch movements (i.e. are straighter and more direct) are more attractive. Furthermore routes that use platforms that are far from the scheduled platform are less desirable. Hence, for each train, the possible routes are ranked to reflect this. When we introduce an additional route for a train, we always add the most preferable route from its set of routes that are not currently considered. Once a route is in the model and its possible paths have been generated, however, it is considered as equally attractive as all other routes for that train currently in the model. The only discriminating factor is the path's deviation. This is, however, likely to be automatically longer for less attractive paths by virtue of the fact that a large number of switch movements typically coincides with a slower speed limit.

5.2.3 Problem Expansion Routine

The initialization step of the solution approach involves constructing t route tree structures, i.e. the planned route for each train. In this paper we only consider disruptions resulting from delayed trains and all initial delays are included in this construction phase. Since each of the route tree structures has several paths, depending on the magnitude of the initial delays, one cannot completely rule out the possibility of finding a set of conflict free paths without enforcing any additional delays. This train path set defines model (5)-(7) and the optimal solution to its LP relaxation is obtained using the method described in Section 5.1. The presence of a null path in this solution indicates the disrupted routing is no longer feasible, and we expand this initial problem using the techniques above.

To determine which trains to provide with extra flexibility, we perform a fractional analysis of the optimal solution by constructing the sum of fractions table F_M. This matrix has dimension $(t \times \tau)$, where there is a row for each train and each column is uniquely associated with one of the variables appearing in the optimal basis B_D of the dual representation for the optimal train path partition. Each element of F_M is given as

$$[F_M]_{ij} = \sum_{k=1}^{\tau} \{[B_D]_{kj} x_k : [T_B]_{ik} = 1\}.$$

The value x_k is directly obtained from the dual value on constraint k. This matrix simply stores the fractional coverage of each tints resource by train path in the optimal train path partition. Using this table one can identify which trains are clashing, during what time interval(s) the clash occurs, and which track section the trains are competing for. Each train involved in a conflict has its maximum number of halting intervals increased by one and receives one additional route. This delay is permitted on track sections prior to the train's first conflicting track section and the additional route received is the one that has the fewest number of track sections in common with the last route added for the train.

Special consideration is given to null paths in the optimal train path partition at value one. For such trains, no information is directly available in F_M as to which trains are preventing the train from passing. One can, however, identify tints resources that are likely to be restricting the movement of the train and identify which trains are currently occupying them. Quite often this occurs when the trains compete for the first track section of their planned route. One can easily work out the minimum separation time required to ensure that the trains can enter the junction without conflict. This minimum time is then used as an entry delay to the junction for the following train, and its tree structures are priced accordingly.

We allow a maximum of four problem expansions and during each one attempt to provide trains with enough flexibility to avoid the current conflicts. We also add to the problem an additional route for trains that, as a result of the disruption, are excessively delayed (although not clashing) in the hope of finding an alternative route with less delay. A train is excessively delayed if it is 150 seconds behind schedule as a result of the disruption. If no null path appears in the optimal train path partition, we immediately begin the branch-and-price routine discussed below. If, on the other hand, at least one null path is in the optimal train path partition after four expansions, we accept that a train may have to be cancelled and use the subsequent branching strategy to identify which one. This is evidence to suggest the disruption is severe enough that manual intervention may be necessary.

5.3 Branch-and-Price Framework

Branching is necessary if the optimal train path partition (post expansion) contains train paths at fractional value. To force integrality we implement a form of constraint branching (see [25]) where the "1-branch" forces a train to claim a particular tints resource, while the "0-branch" prevents a train from claiming a particular tints resource (or any tints resource banned by the tints resource defined in the branch). The branch selection (train, conflicting tints resource) involves constructing the table F_M above and identifying the matrix entry with the largest value. If the selected tints resource is only a banned tints resource for the train (i.e. the train does not actually traverse the associated track section), then the tints resource is replaced by a tints resource that has the same time interval and a track section the train traverses (arbitrarily selected if more than one). Implementation of the branch-and-price is somewhat complicated within our solution approach as optimal train path partitions are only available in their alternative dual representation. One can easily prevent train paths banned by the branch from appearing in the pricing by removing the appropriate arcs; however, removing train paths not satisfying the branch from the optimal train path partition amounts to making currently active constraints of the corresponding dual representation inactive. This can be achieved through a dual implementation of the standard two phase approach of the revised simplex. We omit details here, but refer the reader to [19].

We also impose two further restrictions in the branching process. To speed up the convergence of the restricted problems in the branch-and-price tree, all train paths at value one in the optimal LP train path partition are fixed to one. Furthermore, provided the optimal solution to the LP relaxation of model (5)-(7) contains no null paths at positive value, any node of the branch-and-price tree for which the optimal train path partition contains a null path is fathomed. This is because no feasible completion of the node will have a solution routing all trains. There is no guarantee, of course, that an integer solution routing all trains exists despite the fact that optimal solution to the root node contains no null paths. However, by providing each train with up to 30 minutes of delay on up to four strategically selected additional routes before commencing the branching it seems unlikely, practically speaking, that one will encounter such a situation. The computational results of the subsequent Section confirm this. Note that in this phase of the algorithm no additional route tree structures are constructed and no more delay is permitted.

6 Computational Results

To test the proposed methodology, data on both the rolling stock as well as the infrastructure of a reasonable, yet complex, sized junction in Germany was made available by Deutsche Bahn. A diagram of the junction is given in Figure 5. This junction consists of a main station containing 8 platforms (some of which consist of smaller subplatforms), several smaller stations that each have a couple of platforms, and a freight yard. All possible stopping areas, whether platforms or freight train holding areas, are indicated in red. The junction consists of 524 track sections, contains both one-way and bi-directional track, has two high speed lines for express trains, and can be entered (and/or exited) from one of 10 possible directions (labelled A-J). A high frequency of heterogeneous rail traffic visits the junction daily, and it hence provides a sufficiently complicated, practical sized problem to test or methodology.

We consider two basis timetables from an actual midweek timetable operated by Deutsche Bahn. The first considers a one hour time horizon beginning at 7am, whereas the second is a two hour instance running from 7am-9am. The trains of the first are included in the second. Details of the train type composition can be found in Table 1. As an indication, freight trains can be up to 640 metres in length, regional trains can be up to 150 metres, and express trains are never longer than 360 metres. For the test case, train connections are unknown, and we do not model the shunting movements of a train from its arrival platform to the passenger train holding area if it has an extended dwell time in the station. This is primarily because the necessary data on the infrastructure of the passenger train holding area was unavailable. Trains are hence categorized as inbound trains (arrive at a platform and are moved to the holding area), outbound trains (depart the station after emerging from the holding area), and through trains (leave from the platform they arrive at and remain at the platform throughout the duration of their dwell time in the station). For each timetable 45 scenarios are constructed by randomly delaying 15 sets of a certain number of trains. For the first timetable, instances with 8, 11, and 14 delayed trains are considered and are chosen at random from those trains that arrive within the first 40 minutes. The instances for the

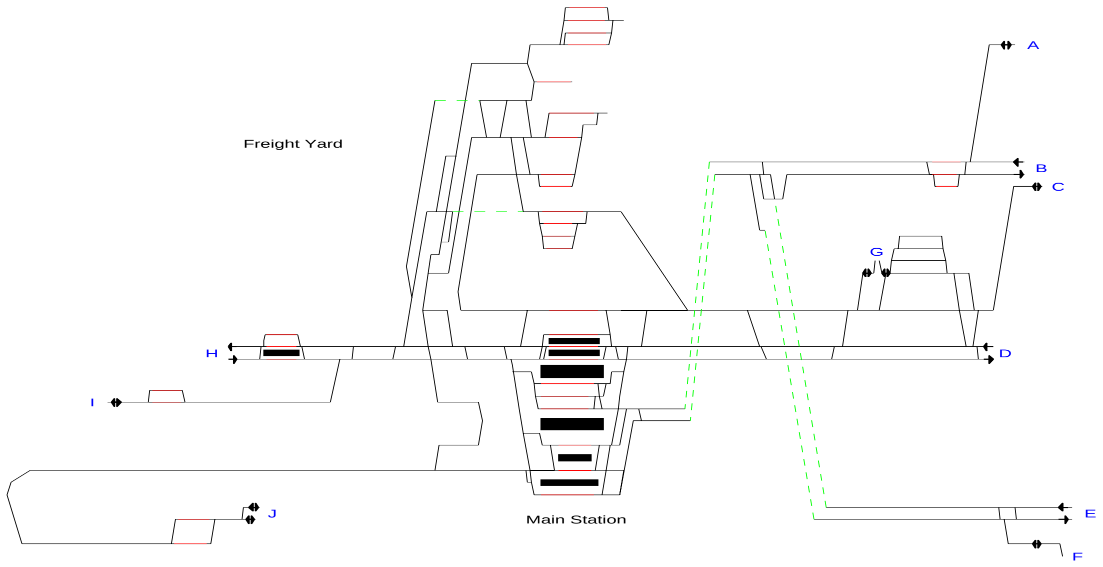

Figure 5: Deutsche Bahn Test Junction

Table 1: Deutsche Bahn Test Instances – Train Types

Instances	#Trains	Freight	Regional	Express	Scenarios
Timetable 1	31	12	15	4	45
Timetable 2	66	23	33	10	45

second timetable consider 10, 14, and 18 delayed trains, and the delayed trains are chosen from those trains that arrive in the first hour. This selection process is an attempt to ensure that the delay propagates. The amount to delay each train by is chosen from a uniform distribution with a lower parameter of 5 minutes and an upper parameter of 15 minutes. Each train has between 2-75 possible routes (consisting of up to 48 track sections) as well as up to three scheduled stops within the junction. As an indication, the largest route tree structure contains in excess of 15,000 labels, and the greatest number of paths in any tree is around 1500. Furthermore, for each instance we consider an unweighted and weighted instance. The unweighted instance sets $\rho_i = 1$ for all trains, while the weighted instance considers different weights for different train types. In particular, $\rho_i = 1$ if train i is a freight train, $\rho_i = 10$ if train i is a regional train, and $\rho_i = 20$ if train i is an express train. The weights have been arbitrarily chosen, and this objective function attempts to delay freight trains in preference to passenger trains.

The results of the biggest instances for each timetable (i.e. 14 delayed trains for the first timetable and 18 delayed trains for the second timetable) are given in Tables 2 and 3. We have omitted the other tables due to space restrictions. They are, however, similar and can be found in [19]. All tests have been performed on an Intel Core 2 Duo, 2.4 GHz computer with 2GB RAM and algorithms are coded in the C++ programming language. A time limit of 270 seconds is imposed on the complete solution approach to reflect the real-time requirement and all weighted instances are marked with an asterisk. For each instance the following information is provided: the total initial (primary) delay (PD), the optimal solution to the LP relaxation (LP), the best integer solution found (IP), the number of nodes evaluated when branching (tn), the gap between the LP and IP solutions, and the CPU time in seconds (t). Other statistics relating to the solutions are also provided. This includes the total number of constraints added to all dual representations (CA), the total number of constraints removed (CR), the optimal basis size of the dual representation of the optimal LP train path partition (OBS), the number of positive train path variables in the optimal train path partition (v), the number of trains that have their route changed (RC), the number of trains that receive a delayed train path (DT), the total number of delayed train paths considered (TDP), the delays to each of the different train types (FD=freight, RD=Regional, and ED=Express), the total delay in restoring feasibility (TD), and the percentage increase of the initial delay incurred in restoring feasibility.

In all tests the delays are severe enough to make the planned routing infeasible. One can observe from Tables 2 and 3, that solutions within a few percent of optimality are obtained, often requiring minimal branching. In some instances, however, the branching process could not be completed within the time limit. For such instances, one can also observe gaps of more than 10% between the LP and IP solutions (see instance 6-14 and 11-18* as examples). Not surprisingly, such instances require the addition and removal of the greatest number of constraints and are also the ones that use the greatest number of delayed paths. Since the branching strategy is heuristic in that train paths at value one in the optimal solution to the root node are fixed, one cannot be sure that the integer solutions reported for those instances that do terminate within 270 seconds are optimal. While gaps of less than 5% are satisfactory, gaps of more than 10% are a cause for concern. To ascertain the quality of such solutions as well as the solutions that are simply the best known solution when the algorithm terminates at the 270 second limit, we use the commercial solver Cplex 10.1 to provide a comparison. The Cplex version of each test instance contains all train paths present in the last expanded problem. That is, Cplex considers the static model (1)-(4) and contains all train paths that are available to be priced in the route tree structures upon completion of the problem expansion routine. This is solved as a MIP model and we use the default settings of Cplex. The results are given in Table 4. The table also reports the total number of paths, the solution found by Cplex, the time taken by Cplex, the solution found using our dual update approach(DA IP), the time at which the last solution was found using our approach (tli), and statistics concerning the required changes to the planned schedule associated with the Cplex solution to recover feasibility. Note that the larger instances approach the memory limitations of the computer and one instance in particular (11-18) could not be solved for this reason. Cplex confirms that all solutions found using our approach are in fact within a few percent of optimality. The worst gap being at most 3.5% (6-14*). This is despite the fact that the lower bound obtained at the root node can be more than 10% away from the optimal solution. However, it does suggest that in many cases our approach was in process of confirming optimality when the 270 second time limit was reached. This is further reinforced by the tli statistic, which suggests that there is usually no improvement in the integer solution once 60 seconds have elapsed.

It is worth mentioning here that our approach does not suffer from the same memory limitations as Cplex. Unlike Cplex, our approach was able to solve instance 11-18. To determine the quality of this solution, a full branch-and-price was implemented (i.e. no variable fixing) and confirmed that this solution was within 2% of optimality. It should also be noted that the Cplex option does not completely mimic our approach; there is no implementation of a problem expansion routine. One can see that it is computationally intractable to a priori generate all combinations of delay and routing possibilities for the trains, and for this reason a dynamic approach is essential.

Tables 2 and 3 also confirm our hypothesis that τ can be expected to be a small multiple of the number of trains. The statistic v shows that the number of positive train path variables in the optimal train path partition is typically a small multiple of the number of trains. Looking at the OBS statistic one can also get a rough indication of the size of any dual representation. A comparison of the different objective functions is also interesting. When train priorities are included, it would appear that one is happy to accept a greater amount of total delay as long as the disruption to the higher priority trains is reduced (see 8-18 for an example). Throughout the tables one can find instances where the unweighted objective function outperforms the weighted

Table 2: Deutsche Bahn 1–hour 14 Train Case

	Solve Statistics					Solution Statistics											
Case	PD(s)	LP	IP	tn	G(%)	t(s)	CA	CR	OBS	v	RC	DT	TDP	FD(s)	RD(s)	ED(s)	TD(s)	P(%)
1–14	8,400	9,383.35	9,412	43	0·31	45·10	2,464	430	187	48	5	2	187	1,754	6,798	860	9,412	12·05
1–14*		84,831.00	85,794	13	1·14	30·97	541	177	205	40	4	4	94	1,834	6,798	799	9,431	12·27
2–14	8,520	9,586.00	9,726	115	1·46	187·78	8,008	1,441	237	49	8	4	407	2,010	6,028	1,688	9,726	14·15
2–14*		93,475.30	94,172	47	0·75	77·47	3,340	766	268	47	9	4	151	2,192	6,028	1,585	9,805	15·08
3–14	9,000	9,161.33	9,266	7	1·14	28·75	639	214	238	40	6	3	122	2,904	6,300	62	9,266	2·96
3–14*		65,515.20	67,144	19	2·49	56·70	1,646	532	221	39	6	3	371	2,904	6,300	62	9,266	2·96
4–14	8,220	8,244.83	8,264	3	0·23	6·46	191	54	146	35	7	1	10	1,510	6,224	530	8,264	0·54
4–14*		72,750.70	73,259	11	0·70	14·64	572	91	149	38	7	1	18	1,749	6,175	488	8,412	2·34
5–14	9,000	10,245.30	10,479	47	2·28	145·31	3,409	807	272	47	4	4	485	1,774	7,844	861	10,479	16·43
5–14*		93,338.00	96,062	47	2·92	251·21	3,943	1,003	250	44	4	4	728	2,082	7,676	861	10,619	17·99
6–14	7,680	8,751.00	9,901	37	13·14	270·92	2,531	860	221	43	5	6	694	2,160	6,439	1,302	9,901	28·92
6–14*		72,879.20	83,831	75	15·03	270·46	4,792	974	226	50	5	6	842	3,041	6,503	788	10,332	34·53
7–14	7,140	7,612.17	7,776	35	2·15	23·21	2,020	320	147	35	2	1	9	1,976	4,971	829	7,776	8·91
7–14*		66,881.70	67,975	17	1·63	13·25	962	157	150	33	3	2	3	2,505	4,971	788	8,264	15·74
8–14	8,760	10,603.10	10,686	9	0·78	17·13	754	267	202	31	6	5	133	1,294	7,543	1,849	10,686	21·99
8–14*		112,875.00	113,704	13	0·73	18·98	988	275	213	31	6	5	151	1,294	7,543	1,849	10,686	21·99
9–14	7,140	7,654.63	7,723	29	0·89	53·86	1,754	151	158	34	5	1	81	2,137	4,918	668	7,723	8·17
9–14*		63,126.10	65,907	21	4·41	22·31	1,440	222	180	38	5	1	128	2,137	5,041	668	7,846	9·89
10–14	7,200	8,208.89	8,515	207	3·73	270·04	13,931	2,155	238	53	10	4	503	1,683	6,404	428	8,515	18·26
10–14*		66,973.30	69,081	35	3·15	169·78	2,614	440	279	42	9	4	277	2,161	5,836	428	8,425	17·01
11–14	7,620	7,625.00	7,625	1	0·00	4·42	65	0	120	31	2	1	11	944	5,972	709	7,625	0·07
11–14*		74,758.00	74,758	1	0·00	4·48	66	0	121	31	2	1	11	968	5,961	709	7,638	0·24
12–14	7,380	8,408.50	8,749	69	4·05	265·41	4,496	728	239	51	6	4	305	2,262	5,676	811	8,749	18·55
12–14*		74,307.30	75,242	9	1·26	36·75	673	180	236	39	7	4	84	2,262	5,676	811	8,749	18·55
13–14	7,620	9,019.78	9,101	11	0·90	27·33	873	170	214	43	3	4	84	1,117	6,579	1,405	9,101	19·44
13–14*		95,003.00	95,003	1	0·00	12·44	151	0	200	31	4	4	42	1,113	6,579	1,405	9,097	19·38
14–14	8,520	9,774.42	10,269	71	5·06	271·19	4,760	1,136	212	47	2	4	636	1,787	7,171	1,311	10,269	20·53
14–14*		93,112.50	98,757	73	6·06	270·89	4,427	1,042	188	42	2	7	725	1,787	8,001	848	10,636	24·84
15–14	8,700	9,544.50	9,616	11	0·75	31·02	726	145	176	45	4	3	99	2,029	5,958	1,629	9,616	10·53
15–14*		92,735.00	94,189	49	1·57	129·70	2,994	618	199	49	5	3	389	2,029	5,958	1,629	9,616	10·53

Table 3: Deutsche Bahn 2–hour 18 Train Case

	Solve Statistics						Solution Statistics											
Case	PD(s)	LP	IP	tn	G(%)	t(s)	CA	CR	OBS	v	RC	DT	TDP	FD(s)	RD(s)	ED(s)	TD(s)	P(%)
1–18	9,240	9,948.00	9,948	1	0·00	16·85	145	0	259	66	8	2	30	2,794	6,661	493	9,948	7·66
1–18*		79,264.00	79,264	1	0·00	17·29	146	0	260	66	8	2	33	2,794	6,661	493	9,948	7·66
2–18	9,960	11,894.00	11,958	11	0·54	32·93	1,419	266	393	72	9	5	100	2,335	7,843	1,780	11,958	20·06
2–18*		115,992.00	116,445	5	0·39	28·25	765	176	395	66	8	5	91	2,415	7,843	1,780	12,038	20·86
3–18	9,480	10,423.00	10,423	3	0·00	15·80	193	106	304	69	6	4	53	2,383	7,487	553	10,423	9·95
3–18*		88,313.00	88,313	3	0·00	15·86	195	109	306	69	6	4	53	2,383	7,487	553	10,423	9·95
4–18	9,780	10,878.00	11,047	9	1·55	87·81	1,285	372	377	69	6	4	247	1,076	8,597	1,374	11,047	12·96
4–18*		110,618.00	112,116	5	1·35	47·13	753	233	332	73	6	4	171	1,076	8,728	1,188	10,992	12·39
5–18	9,720	10,526.00	10,526	1	0·00	16·47	146	0	260	66	6	2	29	1,657	7,836	1,033	10,526	8·29
5–18*		100,677.00	100,677	1	0·00	16·52	146	0	260	66	6	2	29	1,657	7,836	1,033	10,526	8·29
6–18	11,100	12,846.90	13,170	69	2·52	226·91	7,746	1,101	328	79	5	3	780	3,631	8,120	1,419	13,170	18·65
6–18*		112,507.00	112,564	3	0·05	29·56	466	152	363	68	7	3	55	3,744	8,044	1,419	13,207	18·98
7–18	10,260	11,949.80	12,081	39	1·10	35·62	4,233	350	319	77	6	4	105	2,026	8,723	1,332	12,081	17·75
7–18*		114,675.00	115,987	41	1·14	40·48	4,384	358	310	78	4	4	130	2,117	8,723	1,332	12,172	18·64
8–18	10,680	11,155.60	11,276	3	1·08	25·90	494	179	388	71	11	0	59	2,215	7,534	1,527	11,276	5·58
8–18*		106,200.00	107,675	11	1·39	29·41	1,339	251	389	69	11	1	84	2,605	7,637	1,435	11,677	9·34
9–18	10,020	11,671.00	11,674	3	0·03	20·95	304	62	228	68	2	3	36	2,119	7,565	1,990	11,674	16·51
9–18*		117,569.00	117,569	1	0·00	16·05	116	0	227	66	2	3	25	2,119	7,565	1,990	11,674	16·51
10–18	10,680	12,221.00	12,221	1	0·00	18·81	179	0	289	66	10	4	20	1,454	9,927	840	12,221	14·43
10–18*		117,524.00	117,524	1	0·00	19·25	178	0	288	66	10	4	20	1,454	9,927	840	12,221	14·43
11–18	10,200	11,738.30	12,371	43	5·39	274·08	4,667	903	380	85	8	5	643	1,954	8,741	1,676	12,371	21·28
11–18*		113,019.00	125,889	87	11·39	270·55	10,851	2,000	369	80	8	5	1,426	1,939	9,043	1,676	12,658	24·10
12–18	10,680	11,802.00	11,934	3	1·12	33·39	496	171	378	69	4	3	100	630	11,190	114	11,934	11·74
12–18*		113,490.00	114,810	5	1·16	33·04	667	186	341	68	4	3	94	630	11,190	114	11,934	11·74
13–18	11,100	11,964.20	12,481	155	4·32	272·60	16,358	1,369	408	91	12	2	675	2,767	8,981	733	12,481	12·44
13–18*		101,812.00	107,227	149	5·32	270·63	15,719	1,297	415	90	13	2	620	2,767	8,980	733	12,480	12·43
14–18	8,760	10,823.00	11,083	11	2·40	43·48	1,196	151	281	72	7	4	51	1,261	8,546	1,276	11,083	26·52
14–18*		108,698.00	109,702	17	0·92	50·72	1,810	188	287	72	6	4	51	1,722	8,246	1,276	11,244	28·36
15–18	10,740	11,566.50	11,641	11	0·64	28·66	1,226	226	288	74	5	3	99	3,062	8,123	456	11,641	8·39
15–18*		90,833.10	93,412	25	2·84	51·04	2,834	464	297	85	5	3	303	3,062	8,123	456	11,641	8·39

Table 4: Cplex Resolution and Comparison

	Case	Paths	PD(s)	Cplex IP*	$t(s)$	DA IP	$tli(s)$	G(%)	RC	DT	TD(s)
31 Train Instances	4–8	8,489	4,920	6,084	25·59	6,291	144·21	3·40	3	5	6,084
	14–8	23,974	5,340	6,144	80·63	6,144	36·77	0·00	6	3	6,144
	14–8*	25,103	5,340	58,664	90·11	58,664	174·13	0·00	5	3	6,144
	4–11*	4,970	5,280	65,801	11·27	65,801	11·51	0·00	6	1	5,864
	6–11	16,706	6,900	9,179	47·39	9,179	32·60	0·00	2	6	9,179
	6–11*	15,033	6,900	86,767	48·07	88,267	59·95	1·73	1	7	9,179
	7–11	7,606	5,820	6,984	19·56	6,984	42·64	0·00	10	3	6,984
	14–11	11,854	6,540	7,587	41·75	7,619	45·36	0·42	9	4	7,587
	14–11*	22,365	6,540	62,576	88·64	62,576	58·47	0·00	8	4	7,768
	15–11*	30,445	6,300	60,506	104·83	60,506	28·88	0·00	0	2	7,339
	6–14	25,426	7,680	9,742	167·61	9,901	20·44	1·63	5	6	9,742
	6–14*	20,457	7,680	80,981	82·67	83,831	257·28	3·52	6	6	10,182
	10–14	18,101	7,200	8,515	61·53	8,515	105·41	0·00	10	4	8,515
	14–14	19,198	8,520	10,269	64·61	10,269	39·62	0·00	2	4	10,269
	14–14*	17,760	8,520	97,087	58·82	98,757	234·50	1·72	2	5	10,469
66 Train Instances	8–10*	15,395	5,220	62,569	55·52	62,569	37·40	0·00	4	3	6,835
	12–10	18,003	6,240	8,857	69·45	8,857	51·24	0·00	4	4	8,857
	12–10*	18,003	6,240	70,155	63·62	70,155	51·87	0·00	4	5	8,891
	14–10	14,856	6,120	8,199	53·61	8,199	54·58	0·00	8	4	8,199
	14–10*	12,951	6,120	85,358	45·09	85,358	38·96	0·00	7	3	8,685
	1–14*	25,456	7,080	87,209	91·81	87,209	236·89	0·00	6	4	9,094
	2–14*	14,323	7,260	71,055	58·05	72,405	62·59	1·90	6	4	8,947
	5–14*	19,378	8,400	109,500	108·83	109,500	47·80	0·00	10	6	10,781
	13–14	22,742	8,040	10,262	73·80	10,262	50·10	0·00	7	5	10,262
	13–14*	20,434	8,040	91,236	66·17	91,236	38·48	0·00	6	6	10,247
	11–18	31,028	10,200	–	–	12,371	202·77	–	–	–	–
	11–18*	22,468	10,200	125,889	151·55	125,889	113·46	0·00	8	5	12,658
	13–18	22,658	11,100	12,481	88·38	12,481	208·60	0·00	11	2	12,481
	13–18*	24,773	11,100	107,227	102·27	107,227	60·81	0·00	13	2	12,480

objective function. This is because the route tree structures in each problem are dependent on the problem expansion routine and may in fact be different. It is encouraging to see that both tables show the additional delay introduced to recover feasibility is typically small - around 10-30% of the initial delay.

7 Conclusions and Future Work

In this paper we have considered an important operational problem in the railway industry. We have proposed a set packing inspired formulation with a resource based constraint system that implicitly resolves conflicts and developed a branch-and-price approach that exploits the flexibility of the model to be dynamically updated. The solution approach attempts to expand an infeasible, severely restricted problem by identifying conflicting trains and intelligently include extra flexibility without creating an intractable problem. Extra flexibility is provided in the form of delayed train paths or additional routes. A key benefit of the model proposed is that it is identical to that which can be used for the tactical and strategic planning level junction routing problems of timetable feasibility and capacity assessment, even though the solution method is different.

Numerical results on a real life test instance arising in Germany show that the proposed methodology performs well in that it can find solutions within a few percent of optimality in reasonable time while maintaining a delay propagation factor of between 10-30%. Furthermore, the dynamic nature of the approach shows that it is unlikely to suffer from the same limitations as that of an equivalent, static, full formulation solve. Through a comparative study of the solutions obtained using two different objective functions, we have also assessed the impact of including train priorities. The results are consistent with what one would expect in reality and show that by choosing suitable weights one can reduce the delay on higher priority trains by shifting it to less important trains (although this may not be the solution with the least total delay).

References

[1] B. Adenso-Diaź, M. Olivia González, and P. González-Torre. On-line timetable re-scheduling in regional train services. *Transportation Research B*, 33:387 – 398, 1999.

[2] X. Cai and C. J. Goh. A fast heuristic for the train scheduling problem. *Computers and Operations Research*, 21(5):499 – 510, 1994.

[3] X. Cai, C. J. Goh, and A. I. Mees. Greedy heuristics for rapid scheduling of trains on a single track. *IIE Transactions*, 30:481 – 493, 1998.

[4] G. Caimi, D. Burkolter, and T. Herrmann. Finding delay-tolerant train routings through stations. In H. Fleuren, editor, *Operations Research Proceedings 2004: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR) Jointly Organized with the Netherlands Society*, pages 136–143. Springer Verlag, 2005.

[5] J. Clausen, A. Larsen, J. Larsen, and N. Rezanova. Disruption management. *ORMS Today*, 28(3):40–43, 2001.

[6] J. Clausen, A. Larsen, J. Larsen, and N. Rezanova. Disruption management in the airline industy – concepts, models and methods. Computers and Operations Research, forthcoming, doi:10.1016/j.cor.2009.03.27, 2009.

[7] J. Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32(4):380 – 400, 1998.

[8] I. Şahin. Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research Part B*, 33:511 – 534, 1999.

[9] A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real time. *Transportation Science*, 42(4):405 – 419, 2008.

[10] A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch-and-bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183:643 – 657, 2007.

[11] DB. AG.Daten und Fakten zum Geschäftsbericht, 2005.

[12] X. Delorme. *Modélisation et résolution de problèmes liés à l'exploitation d'infrastructures ferroviaires*. PhD thesis, Université de Valenciennes et du Hainaut Cambrésis, 2003.

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

[14] T. M. Herrman. *Stability of Timetables and Train Routings Through Station Regions*. PhD thesis, Swiss Federal Insitute of Technology Zurich, 2006.

[15] A. Higgins, E. Kozan, and L. Ferreira. Optimal scheduling of trains on a single line track. *Transportation Research Part B*, 30B(2):147 – 161, 1996.

[16] A. Higgins, E. Kozan, and L. Ferreira. Heuristic techniques for single line train scheduling. *Journal of Heuristics*, 3:43 – 62, 1997.

[17] N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourine. Airline disruption management – perspectives, experiences and outlook. *Journal of Air Transportation*, 13:149 – 162, 2007.

[18] Q. Lu, M. Dessouky, and R. C. Leachman. Modeling train movements through complex rail networks. *ACM Transactions on Modeling and Computer Simulation*, 14(1):48 – 75, 2004.

[19] R. M. Lusby. *Optimization Methods for Routing Trains Through Railway Junctions*. PhD thesis, The University of Auckland, 2008.

[20] R. M. Lusby, J. Larsen, M. Ehrgott, and D. Ryan. Railway track allocation: Models and methods. Technical Report 3, Department of Management Engineering, The Technical University of Denmark, 2009.

[21] O. Oliveira and B. M. Smith. A job shop scheduling model for for the single track-railway timetabling problem. Technical Report 2000.21, University of Leeds, 2000.

[22] J. Pachl. *Systemtechnik des Schienenverkehrs*. B.G. Teubner, 2004. In German.

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

[24] J. Rodriguez and L. Kermad. Constraint programming for real-time train circulation management problems in railway nodes. In *Proceedings of the International Conference on Computer Aided Design, Manufacture and Operation in The Railway and Other Advanced Mass Transit Systems*, pages pp. 597–606, 1998.

[25] D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer Scheduling of Public Transport*, pages 269 – 280. North-Holland Publishing Company, 1981.

[26] J. Törnquist. Computer-based decision support for railway traffic scheduling and dispatching: a review of models and algorithms. In *ATMOS2005 (Algorithmic MeThods and Models for Optimization of RailwayS)*, Palma de Mallorca,Spain, October 2005. Dagstuhl Research Online Publication Server (DROPS).

[27] J. Törnquist and J. A. Persson. N-tracked railway traffic rescheduling during disturbances. *Transportation Research Part B*, 41(3):342 – 362, 2007.

[28] G. Yu and Q. Xi. *Disruption Management:Framework, Model and Applications*. World Scientific Publishing Co. Pte. Ltd., Singapore, 2004.

[29] P. J. Zwaneveld, L. G. Kroon, H. E. Romeijn, M. Salomon, S. Dauzere-Peres, S. P. M. van Hoesel, and H. W. Ambergen. Routing trains through railway stations: Model formulation and algorithms. *Transportation Science*, 30(3):181 – 194, 1996.

[30] P. J. Zwaneveld, L. G. Kroon, and S. P. M. van Hoesel. Routing trains through a railway station based on a node packing model. *European Journal of Operational Research*, 128:14 – 33, 2001.

Efficiently coordinating the often large number of interdependent, timetabled train movements on a railway junction, while satisfying a number of operational requirements, is one of the most important problems faced by a railway company. The most critical variant of the problem arises on a daily basis at major railway junctions where disruptions to rail traffic make the planned schedule/routing infeasible and rolling stock planners are forced to reschedule/re-route trains in order to recover feasibility. The dynamic nature of the problem means that good solutions must be obtained quickly. In this paper we describe a set packing inspired formulation of this problem and develop a branch-and-price based solution approach.

A real life test instance arising in Germany and supplied by the major German railway company, Deutsche Bahn, indicates the efficiency of the proposed approach by confirming that practical problems can be solved to within a few percent of optimality in reasonable time.

ISBN 978-87-90855-59-8

DTU Management Engineering
Department of Management Engineering
Technical University of Denmark

Produktionstorvet
Building 424
2800 Kongens Lyngby
Tel. 45 25 48 00
Fax 45 93 34 35

www.man.dtu.dk