Strong Relaxations for the Train Timetabling Problem Using Connected Configurations

Frank Fischer[1] **and Thomas Schlechte**[2]

1 **University of Kassel, Kassel, Germany**
`frank.fischer@mathematik.uni-kassel.de`
2 **LBW Optimization GmbH, Berlin, Germany**
`schlechte@lbw-optimization.de`

Abstract

The task of the *train timetabling problem* or *track allocation problem* is to find conflict free schedules for a set of trains with predefined routes in a railway network. Especially for non-periodic instances models based on time expanded networks are often used. Unfortunately, the linear programming relaxation of these models is often extremely weak because these models do not describe combinatorial relations like overtaking possibilities very well. In this paper we extend the model by so called connected configuration subproblems. These subproblems perfectly describe feasible schedules of a small subset of trains (2-3) on consecutive track segments. In a Lagrangian relaxation approach we solve several of these subproblems together in order to produce solutions which consist of combinatorially compatible schedules along the track segments. The computational results on a mostly single track corridor taken from the INFORMS RAS Problem Solving Competition 2012 data indicate that our new solution approach is rather strong. Indeed, for this instance the solution of the Lagrangian relaxation is already integral.

1998 ACM Subject Classification G.1.6 [Numerical Analysis] Optimization

Keywords and phrases combinatorial optimization, train timetabling, Lagrangian relaxation, connected configurations

Digital Object Identifier 10.4230/OASIcs.ATMOS.2017.11

1 Introduction

The *train timetabling problem* (TTP) or *track allocation problem* aims for determining schedules for a set of trains in a railway network such that certain technical constraints are satisfied. The authors of [7] provide a recent overview of railway timetable design in practice and the combinatorial optimisation models that have been proposed for this application. They identify and compare two major streams of combinatorial optimisation models for railway timetabling – periodic timetabling based on the classic Periodic Event Scheduling Problem (PESP) and aperiodic train timetabling. The problem of timetabling is closely related to the more operational task of train dispatching, see [5] and [19]. In these applications models based on disjunctive formulations are successfully in use which where introduced for classic job scheduling in [21].

In this work, we focus on the non-periodic case which goes back to the seminal paper of [2]. The authors introduced a time index model which is the basis for a vast amount of literature in which this model was used or extended in various aspects or applications, see the following selection [1, 4, 6, 8, 9].

We investigate the time index model variant of the train timetabling model and in particular improve the quality of the relaxation. Our model is based on configuration

17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017).
Editors: Gianlorenzo D'Angelo and Twan Dollevoet; Article No. 11; pp. 11:1–11:16
Open Access Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

networks introduced in [1], which model valid orderings and sequences of train on a single track in the network. A similar approach describing feasible configurations using comparability graphs for train schedules has been investigated in [3]. Extending first ideas from [12], the new approach is to represent combinatorial properties like overtaking possibilities explicitly in the model. Unlike [12], which only handled track segments without overtaking possibility, this includes cases where a limited siding capacity is available along a sequence of tracks.

In a Lagrangian relaxation approach the TTP is solved exactly on small parts of the network (one or two infrastructure arcs) for small subsets of trains (up to three). By carefully solving the subproblems not in isolation but together combinatorial compatibility is guaranteed along a sequence of tracks. We will illustrate the power of the new model and solution approach in some preliminary but very encouraging numerical tests. Indeed, our Lagrangian relaxation approach is able to compute an integer solution for a small but challenging instance of 14 trains on a corridor consisting of single and double track parts and few additional overtaking places.

The paper is organised as follows. Section 2 formally introduces the TTP. Section 3 recapitulates the classic integer programming model based on a time indexed graph formulation and the configuration based models. Section 4 explains the entire solution approach based on Lagrangian relaxation. We will introduce the concept of strong connected configurations in Section 5 in order to model orderings of small subsets of trains for critical parts of the network. The computational experiments are presented in Section 6 showing that the new model and solution approach are clearly much stronger than the classic models. Finally, we conclude and give suggestions for future work in Section 7.

Definitions and Notations

In this paper we use the following notation. We mainly use directed graphs $G = (V, A)$, which may contain loops $(u, u) \in A$. For a directed arc $a = (u, v) \in A$ we write uv and the arc running in the opposite direction is $\overleftarrow{a} = (v, u)$ (if it exists). Given a subset of the nodes $V' \subseteq V$, then $G[V']$ denotes the subgraph induced by V'. Let $x, y \in \mathbb{R}^n$ be two vectors, then the inner product is $\langle x, y\rangle := \sum_{i=1}^{n} x_i y_i$. For a subset of the indices $I \subseteq \{1, \dots, n\}$ the vector x_I refers to the components of x corresponding to I.

2 The Train Timetabling Problem

In this section we describe the *Train Timetabling Problem* (TTP). We focus with our description on the aspects that we need in our subsequent analysis and ignore many properties that have been considered in the literature, in particular for practical applications.

We consider an infrastructure (railway) network $G^I = (V^I, A^I)$. The nodes V^I correspond to stations, track switches or crossings in the network and the arcs A^I correspond to physical tracks or track sections. If trains might stop and wait at some node $v \in V^I$ (e. g. if v is a station), then $vv \in A^I$, otherwise, if trains must not wait at v (e. g. if v is a crossing), then $vv \notin A^I$. Next we have a finite set of trains R with predefined and fixed routes $G^r = (V^r, A^r) \subseteq G^I$, $r \in R$, i. e. G^r corresponds to a path in G^I with the exception that it might contain loops $vv \in A^r$ if train r is allowed to wait at v. Each train has a *starting time* $\vec{t}^r \geq 0$ and the time it takes for train r to traverse arc $a \in A^r$ is called its *running time* $\vec{t}^r_a \in \mathbb{N}_0$ (in minutes). If $a = vv \in V^r$, i. e. if a is a loop corresponding to waiting at v, we set $\vec{t}^r_a := 1$.

Considering all trains running in the network together, certain technical and operational constraints must be satisfied. We deal with two main conditions, station capacities and

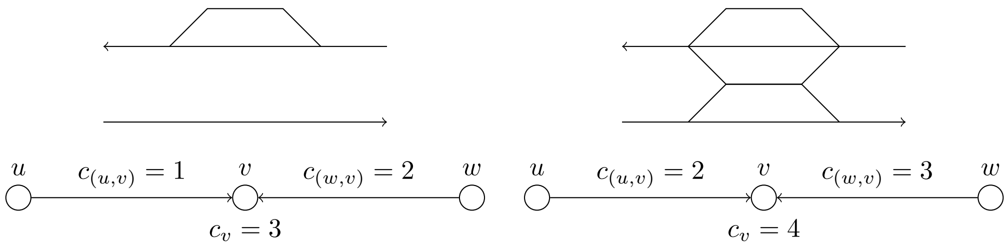

Figure 1 Small stations with absolute and directional capacities.

headway times. First, each node is assigned an *absolute capacity* and each arc is assigned a *directional capacity* $c\colon V^I \cup A^I \to \mathbb{N}$. The absolute capacity c_v for a node $v \in V^I$ is the maximal total number of trains that may visit v at the same time. The directional capacity c_a for an arc $a = (u,v) \in A^I$ is the total number of trains reaching v over a that may be at v at the same time. These two capacities suffice to model small stations and overtaking places quite accurately, see Figure 1.

Second, for each pair of trains $r, r' \in R$ using the same infrastructure arc $a \in A^I$, $a \in A^r \cap A^{r'}$, there is a *minimal headway time* $\tilde{t}^a_{r,r'} \in \mathbb{N}_0$, which is the minimal time (in minutes) that train r' must wait before entering arc a after r has entered a. Note that headway times may also exist for trains running in opposite directions if a corresponds to a physical single line track. In slight abuse of notation we use for this case $\tilde{t}^a_{r,r'}$, too, although $a = (u,v) \in A^r$ and $\overleftarrow{a} := (v,u) \in A^{r'}$. We will assume throughout the paper that headway times satisfy the triangle inequality $\tilde{t}^a_{r,r'} + \tilde{t}^a_{r',r''} \geq \tilde{t}^a_{r,r''}$, see [20] for an in-depth analysis of headway times.

The goal of the TTP is to determine precise arrival and departure times for each train at each of its stations so that all trains run “as early as possible” (we deliberately do not make this more specific at this point) such that the compound schedule satisfies the capacity and headway restrictions.

3 The Model for the TTP

Several models for the TTP have been proposed in the literature. We use one successful approach that is based on time discretised and time expanded networks. Let $\mathcal{T} = \{1, 2, \ldots\} = \mathbb{N}$ be the set of discretised time steps (e. g. in minutes). Then $G^r_{\mathcal{T}} = (V^r_{\mathcal{T}}, A^r_{\mathcal{T}})$ is the associated time expanded graph with (see Figure 2)

$$V^r_{\mathcal{T}} := V^r \times \mathcal{T}, \quad A^r_{\mathcal{T}} := \left\{((u,t),(v,t+\vec{t}^r_{uv}))\colon uv \in A^r, t \in \mathcal{T}\right\}.$$

A feasible *train run* or *train schedule* is a path $P^r \subseteq G^r_{\mathcal{T}}$ starting in a node corresponding to the first station (at some arbitrary time $t \in \mathcal{T}$) and ending in a node corresponding to the last station (at some arbitrary time). We denote the set of all feasible schedules by $\mathcal{P}^r$. In order to model the TTP as a mixed integer program (MIP) we introduce binary variables $x^r\colon A^r_{\mathcal{T}} \to \{0,1\}$, $r \in R$, with the usual interpretation $x^r_a = 1 \iff a \in P^r \in \mathcal{P}^r$ if P^r is the path (schedule) selected for r. With a slight abuse of notation we write $x^r \in \mathcal{P}^r$ if x^r is the incidence vector of a path $P^r \in \mathcal{P}^r$.

Next we model the capacity and headway constraints. For a given infrastructure arc $a \in A^I$ we denote by $R^a := \left\{r \in R\colon a \in A^I\right\}$ the set of trains running on this arc. The

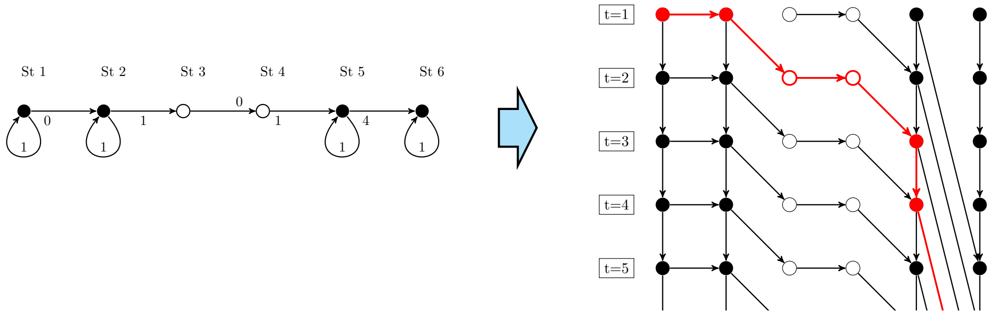

Figure 2 Train graph G^r and its time expansion $G^r_{\mathcal{T}}$. Nodes at which r may wait have a loop. The red path corresponds to the train's timetable.

absolute and directional capacity constraints can be formulated as follows:

$$\sum_{\substack{r \in R \\ a=((u,t_u),(v,t)) \in A^r_{\mathcal{T}}}} x^r_a \le c_v, \qquad v \in V^I, t \in \mathcal{T}, \tag{1a}$$

$$\sum_{\substack{r \in R^{uv} \\ a=((u,t_u),(v,t)) \in A^r_{\mathcal{T}}}} x^r_a \le c_{uv}, \qquad uv \in A^I, u \neq v, t \in \mathcal{T}. \tag{1b}$$

Note that the left-hand side of (1b) sums only over the (arcs of the) trains R^{uv} actually using the arc $uv \in A^I$, whereas (1a) sums over all trains arriving at $v \in V^I$ independent of the source node.

In order to model headway constraints we use the idea of configurations suggested in [1]. First observe that two train run arcs $((u,t_u),(v,t_v)) \in A^r$ and $((u,t'_u),(v,t'_v)) \in A^{r'}$ with $u \neq v$ violate the headway restrictions on infrastructure arc $a = (u,v) \in A_I$ if and only if

$$-\tilde{t}^a_{r,r'} < t_u - t'_u < \tilde{t}^a_{r',r}, \tag{2}$$

(i. e. if the departure times at u are too close). This allows to define a *conflict graph* $G_{\text{hw}} = (V_{\text{hw}}, E_{\text{hw}})$ on the run arcs, i. e.

$$\begin{aligned} V_{\text{hw}} &:= \{(a,r) \colon a = ((u,t_u),(v,t_v)) \in A^r, u \neq v, r \in R\}, \\ E_{\text{hw}} &:= \Big\{ \{(a,r),(a',r')\} \colon a = ((u,t_u),(v,t_v)) \in A^r, a' = ((u,t'_u),(v,t'_v)) \in A^{r'}, \\ &\qquad\qquad u \neq v \text{ and } r = r' \text{ or } a, a' \text{ satisfy } (2) \Big\}. \end{aligned}$$

Note that each train $r \in R$ can run only once on each infrastructure arc, so two different runs $(a,r),(a',r)$ of the same train (on the same infrastructure arc) are implicitly in conflict. The configuration model suggested by the authors in [1] works by enforcing the headway restrictions independently on each infrastructure arc. For this, let $uv \in A^I$, $u \neq v$, and denote by

$$V^{uv}_{\text{hw}} = \{(a,r) \in V_{\text{hw}} \colon a = ((u,t_u),(v,t_v)) \in A^r_{\mathcal{T}}, r \in R\}$$

the set of all run arcs corresponding to some train running on uv. A *configuration* is the characteristic vector of some set of conflict free train runs on uv. In other words, a configuration corresponds to a stable set in the conflict graph $G^{uv}_{\text{hw}} := G_{\text{hw}}[V^{uv}_{\text{hw}}]$. We set

$$\mathcal{C}^{uv} := \{x \colon V^{uv}_{\text{hw}} \to \{0,1\} \colon x \text{ is a characteristic vector of a stable set in } G^{uv}_{\text{hw}}\}.$$

In order to use configurations in the model, we introduce *configuration variables* $z^{uv}: V^{uv}_{\text{hw}} \to \{0,1\}$, $uv \in A^I$, and coupling *configuration constraints*

$$x^r_a = z^{uv}_{(a,r)}, \qquad a = ((u,t_u),(v,t_v)) \in A^r_{\mathcal{T}}, r \in R, uv \in A^I. \tag{3}$$

Constraints (3) couple the train graphs with the configurations and ensure that a train may run on an arc $a \in A^r_{\mathcal{T}}$ if and only if that arc is allowed in the selected configuration.

The objective function is very simple and taken from [11]. For a train run arc $a = ((u,t_u),(v,t_v)) \in A^r_{\mathcal{T}}$, $u \neq v$, $r \in R$, let $\underline{t}^v$ be the earliest possible arrival time of r at v and $\ell^r_{uv} := \vec{t}^r_{uv} / \sum_{a \in A^r} \vec{t}^r_a$ the relative length (running time) of arc uv compared to the complete train path. Let $\delta > 0$ be the discretisation step size (in seconds) and α_r, $r \in R$, be some specific weight factor $\alpha_r > 0$. The cost of arc $a = ((u,t_u),(v,t_v)) \in A^r_{\mathcal{T}}$ is

$$w^r_a = \begin{cases} \alpha_r \cdot \ell^r_{uv} \cdot (\delta \cdot (t_v - \underline{t}^r_v))^2, & u \neq v, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

These costs penalise a late arrival of a train relative to the earliest possible arrival time quadratically at each node the train visits. So each train aims to run as early as possible with large delay of one train being more expensive than small delays distributed to several trains. Note that the structure of this cost function allows to use the dynamic graph generation technique introduced in [14]. In particular, there is no need for an a-priori bounded maximal time index because the network grows automatically as required during the solution process.

Finally, the *configuration formulation* of the TTP is

$$\text{minimise} \quad \left\{ \sum_{r \in R} \langle w^r, x^r \rangle : x^r \in \mathcal{P}^r, r \in R, z^a \in \mathcal{C}^a, a \in A^I, (1), (3) \right\}. \tag{TTP-cfg}$$

Constraints $x^r \in \mathcal{P}^r$ correspond to paths in $G^r_{\mathcal{T}}$, $r \in R$, and can be formulated by linear flow constraints. Constraints $z^a \in \mathcal{C}^a$, $a \in A^I$, are much more challenging, because they correspond to a stable set polytope in G^a_{hw}, $a \in A^I$. The authors in [1] used a weaker formulation via so called *configuration networks*, which can also be described by linear flow constraints. The model (TTP-cfg) therefore gives rise to a MIP formulation for the TTP, which could, in principle, be solved by state-of-the-art solvers. This approach, however, is computationally intractable even for small instances. Therefore, we use another solution approach to be described in the next section.

▶ Remark. Another typical formulation (see, e. g. [11]) can be obtained by replacing configurations and constraints (3) by inequalities of the form $x^r_a + x^{r'}_{a'} \leq 1$ for all $\{(a,r),(a',r')\} \in E_{\text{hw}}$. These inequalities forbid the use of train run arcs that violate headway restrictions. However, this formulation is often even weaker than the configuration formulation [1] and will not be discussed in this paper.

4 Solution Approach

Standard solution approaches for (TTP-cfg) are usually based on column generation or Lagrangian relaxation, see, e. g. [1]. In this work we focus on Lagrangian relaxation, which we used in all of our tests.

Lagrangian relaxation is based on the following observation: without the coupling constraints (1) and (3), problem (TTP-cfg) decomposes into smaller, independent subproblems. Therefore, instead of enforcing these constraints their violation is penalised in the objective function. Write the coupling constraints (1) as $\sum_{r\in R} M^r_{\text{cap}} x^r \le c$ for appropriate matrices M^r_{cap}, $r \in R$. Then the Lagrangian dual problem is

$$\max_{\substack{y_{\text{cap}}\ge 0\\ y_{\text{cfg}}\text{ free}}} \left[-c^T y_{\text{cap}} + \sum_{r\in R} \min_{x^r\in\mathcal{P}^r} \left\langle w^r + (M^r_{\text{cap}})^T y_{\text{cap}} + y_{\text{cfg}}, x^r \right\rangle + \sum_{a\in A^I} \min_{z^a\in\mathcal{C}^a} \left\langle -y_{\text{cfg}}, z^a \right\rangle \right].$$

An optimal solution of this problem gives a lower bound on the optimal objective value of the primal problem (TTP-cfg). Because the minimum over affine functions is concave and the sum of concave functions is concave as well, the term within the max is a concave function in $(y_{\text{cap}}, y_{\text{cfg}})$. Hence (by taking its negative) the above problem is a convex optimisation problem that can be solved using, e. g. subgradient or bundle methods. In our implementation we use a proximal bundle method [17] that in addition to optimal multipliers $y_{\text{cap}}, y_{\text{cfg}}$ computes approximate (fractional) primal solutions x^r, $r \in R$. These algorithms require the solution of the subproblems

$$\min_{x^r\in P^r} \left\langle w^r + (M^r_{\text{cap}})^T y_{\text{cap}} + y_{\text{cfg}}, x^r \right\rangle, \qquad r \in R, \tag{5}$$

$$\min_{z^a\in\mathcal{C}^a} \left\langle -y_{\text{cfg}}, z^a \right\rangle, \qquad a \in A^I, \tag{6}$$

for given Lagrangian multipliers $y_{\text{cap}}, y_{\text{cfg}}$. The exact algorithmic details of this solution approach are not important in this paper, as we focus on the modelling aspect. We refer the reader to [11] for more information.

Note that solving the subproblems (5) is easy (because they are shortest-path problems in acyclic networks, see [11]), but solving (6) corresponds to solving a weighted stable-set problem in G^a_{hw}, which is hard in general. However, if the number of trains is sufficiently small, the subproblems (6) can be solved efficiently. This and the observation, that in many instances typically only few trains compete for the same physical track at the same time, motivate the strong configuration approach to be described in the next section.

5 Strong Configurations for Small Subsets of Trains

One important step in solving (TTP-cfg) is the computation of lower bounds, e. g. by the Lagrangian relaxation approach sketched in the previous section. The quality of these bounds has a major impact on the performance and the solution quality of the overall solution process. However, even if configuration subproblems (6) are solved exactly, optimal solutions of the relaxation are very weak, even for trains running on a corridor. Figure 3 shows a typical situation: two trains run on a sequence of single line tracks, such that no overtaking is allowed on the intermediate nodes. For instance, the intermediate nodes could be small local stations without overtaking/passing possibility or the arcs represent a sequence of blocking areas (guarded by signals) which must not be occupied by more than one train. In particular, stopping and waiting at these intermediate nodes is allowed. Obviously, because there is no overtaking possibility, in a feasible solution one train must go first through the complete sequence of tracks and then the other. However, the fractional solution can easily exploit the weak formulation: both trains can run fractionally in short succession and pass at an intermediate node.

The reason for this behaviour is that the combinatorial properties of the network (no overtaking is possible) are not represented well in the current linear model. Headway constraints are only enforced on each single infrastructure arc in isolation but the relation of neighbouring arcs is only enforced by the integrality, which, however, is lost when solving

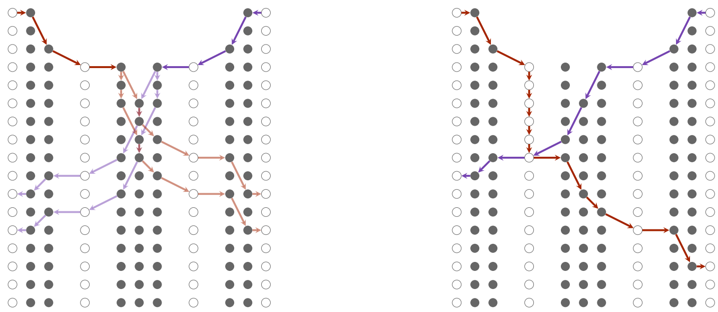

Figure 3 Tiny example with two trains running in opposite directions on a single line track with two passing possibilities. The white nodes have capacity 2, the other nodes have capacity 1. Nodes in the same row correspond to a sequence of stations at the same time step, nodes in the same column correspond to one station but at different time steps (time grows from top to bottom). All arcs are single track. The left picture shows an optimal solution for the standard linear relaxation. Note that the solution exploits the weak formulation allowing the two trains to meet and pass on a single line part. The right picture shows the optimal solution if the strong configurations of Section 5.3 are used: one train has to wait at a capacity 2 node for the other train to pass.

a relaxation. In [12] an approach has been presented that adds some of these relations on track sections without overtaking to the model in terms of so called "ordering constraints". This works by enriching the configuration subproblems by additional variables and coupling them by special equality constraints that enforce that all trains run in the same order on each arc of the track section. Unfortunately, this approach cannot easily be extended to more complex situations.

Thus, we present in this paper an alternative, more direct way of adding combinatorial relations to the *relaxed* models. We will apply this approach to the basic case of no overtaking possibilities (the case the ordering constraint approach has been designed for) in Section 5.3 and show how it can easily be extended to more general settings, which we demonstrate in Section 5.4.

5.1 Connected Configurations

Motivated by the example shown in Figure 3, the goal of our approach is to prevent incompatible configurations that might appear if the subproblems are evaluated in isolation. Whereas in [12] the idea was to use additional constraints that forbid incompatible configurations, the new idea is more direct. When evaluating the subproblems (6) for certain adjacent infrastructure arcs, we *force* the generated configurations to be compatible during the solution of the subproblems. This requires that the subproblems are not evaluated in isolation but some of them must be evaluated together. The downside of this approach is that it leads to much more difficult subproblems so that a careful design of the subproblems and the solution process is mandatory.

Let $P^I = v_0 P_1^I v_1 \dots P_k^I v_k$, $v_i \in V^I$, be a path in the infrastructure network consisting of subpaths P_i^I, $i = 1, \dots, k$. Consider a subset $R' \subseteq R$ of trains running over the complete path P^I. Furthermore, for $i = 1, \dots, k$ we denote by $\pi_i^{\text{in}}, \pi_i^{\text{out}} \colon \{1, \dots, |R'|\} \to R'$ two orderings of the trains R' entering and leaving path P_i^I at v_{i-1} and v_i, respectively. The idea is to solve the TTP *exactly* on each subpath P_i^I for trains R' such that the trains enter P_i^I in order π_i^{in} at v_{i-1} and leave P_i^I with order π_i^{out} at v_i. Thus, the exact solutions for each P_i^I are connected to a solution on the entire path P^I by selecting solutions of compatible orders only, i. e. enforcing

$$\pi_{i-1}^{\text{out}} = \pi_i^{\text{in}}, i = 1, \ldots, k. \tag{7}$$

Formally, for a given path P^I and trains R' let $A_{R'}^I[P^I]$ be the set of those arcs in $\bigcup_{r \in R'} A_{\mathcal{T}}^r$ that correspond to P^I. Then we denote the set of *extended configurations*

$$\mathcal{C}_{\pi^{\text{in}},\pi^{\text{out}}}^{P^I,R'} := \Big\{ x \colon A_{R'}^I[P^I] \to \{0,1\} \colon x \text{ is a characteristic vector of a feasible timetable for } R' \text{ on } P^I \text{ with orders } \pi^{\text{in}},\ \pi^{\text{out}} \Big\}. \tag{8}$$

This set consists of all vectors corresponding to feasible timetables for R' on P^I where the trains run in orders π^{in} and π^{out} when entering and leaving P^I, respectively. In other words, each vector in $\mathcal{C}_{\pi^{\text{in}},\pi^{\text{out}}}^{P^I,R'}$ is a feasible solution when restricting the TTP to only scheduling trains R' on path P^I. Note, in contrast to the standard configurations $\mathcal{C}^a$, the extended configurations may span more than one infrastructure arc, but the solutions are restricted to the fixed orders $\pi^{\text{in}}, \pi^{\text{out}}$. Given linear objective terms $\tilde{c}_i, i = 1, \ldots, k$, we define the *connected configurations subproblem* as

$$\begin{aligned} \text{minimise} \quad & \sum_{i=1}^{k} \min \Big\{ \langle \tilde{c}_i, x \rangle : x \in \mathcal{C}_{\pi_i^{\text{in}},\pi_i^{\text{out}}}^{P_i^I,R'} \Big\} \\ \text{subject to} \quad & \pi_{i-1}^{\text{out}} = \pi_i^{\text{in}}, i = 1, \ldots, k, \\ & \pi_i^{\text{in}}, \pi_i^{\text{out}} \text{ orders of } R'. \end{aligned} \tag{9}$$

Problem (9) selects feasible configurations on each subpath P_i^I in such a way that the order of trains on consecutive subpaths must match. Note, however, that solutions of (9) are *not* feasible TTP solutions on the whole path P^I in general: although the orders of trains on P_i^I and P_{i+1}^I must be the same, the exact times when the trains leave P_i^I may not match the exact times when the trains enter P_{i+1}^I. We deliberately allow this inexactness in the subproblem in order to keep (9) computationally tractable (because really solving the TTP exactly on the whole path P^I becomes very hard quickly).

5.2 Solving Connected Configuration Subproblems

In general, the connected configurations subproblem is computationally hard, in particular, enforcing the compatibility constraints (7). However, if the number of trains R' is sufficiently small (e. g. $|R'| \leq 5$), we can solve it by full enumeration over all possible train orders. First, consider the extended configuration subproblem on some subpath P_i^I for fixed orders $\pi_i^{\text{in}}, \pi_i^{\text{out}}$:

$$v_{i,i}(\pi_i^{\text{in}}, \pi_i^{\text{out}}, \tilde{c}_i) := \min \Big\{ \langle \tilde{c}_i, x \rangle : x \in \mathcal{C}_{\pi_i^{\text{in}},\pi_i^{\text{out}}}^{P_i^I,R'} \Big\}. \tag{10}$$

This subproblem can be modelled and solved straight forward as MIP (it is basically a standard TTP model on a very small network, additionally with fixed train orders). Now denote by

$$\begin{aligned} v_{i_1,i_2}(\pi^{\text{in}}, \pi^{\text{out}}) := \quad \text{minimise} \quad & \min \sum_{i=i_1}^{i_2} \Big\{ \langle \tilde{c}_i, x \rangle : x \in \mathcal{C}_{\pi_i^{\text{in}},\pi_i^{\text{out}}}^{P_i^I,R'} \Big\} \\ \text{subject to} \quad & \pi_{i-1}^{\text{out}} = \pi_i^{\text{in}}, \qquad i = i_1 + 1, \ldots, i_2, \\ & \pi_{i_1}^{\text{in}} = \pi^{\text{in}}, \pi_{i_2}^{\text{out}} = \pi^{\text{out}}. \end{aligned}$$

the optimal value of the connected configuration subproblem restricted to the subpath between v_{i_1} and v_{i_2} with fixed first and last order. Then the following simple recursion holds for $i_1 < i_2$

$$v_{i_1,i_2}(\pi^{\text{in}}, \pi^{\text{out}}) = \min_{\pi \text{ order of } R'} \left[v_{i_1,i_1}(\pi^{\text{in}}, \pi) + v_{i_1+1,i_2}(\pi, \pi^{\text{out}}) \right] \tag{11}$$

and the optimal value of (9) can be computed with dynamic programming:

Algorithm 1 Solve connected configuration subproblem

```
1 Input: P_i^I, c̃_i, i = 1,...,k
2 for i := 1 to k do
3   foreach π_i^in, π_i^out do
4     compute v_{i,i}(π_i^in, π_i^out) by solving (10)
5   end
6 end
7 compute an optimal solution of (9) by (11)
```

For each i the loop in lines 3–5 is executed at most $(|R'|!)^2$ times, i. e. we need to solve at most $k \cdot (|R'|!)^2$ (independent) subproblems (10). The dynamic programming step in line 7 is extremely quick (less than 1 ms) because the recursion has only a depth of k and in each step only $(|R'|!)^2$ numbers have to be considered (recall we assume $|R'| \leq 5$).

5.3 Configurations on Non-Overtaking Sections

In this section we consider the special case of an infrastructure path P^I without overtaking possibility. This implies that the order of trains must not change along the whole path. Therefore, we decompose $P^I = v_0 a_1 v_1 \dots a_k v_k$ with $a_i \in A^I$ into single-arc subpaths $P_i^I = v_{i-1} a_i v_i$, $i = 1, \dots, k$. Obviously, overtaking on one of the subpaths P_i^I is not allowed, hence the extended configuration subproblems (10) have only a feasible solution if $\pi_i^{\text{in}} = \pi_i^{\text{out}}$. Note that this reduces the number of problems to be solved for P_i^I from $(|R'|!)^2$ to $|R'|!$. Furthermore, because no two trains can change their order along P^I, intuitively it might be enough to enforce correctness for each pair of trains only. Hence, we only consider subsets of exactly two trains $|R'| = 2$, reducing the number of subproblems to be solved on each subpath P_i^I to two. Therefore, in order to solve (9) we only need to solve k subproblems on a single arc for each of the two possible orders of the trains and take the better one of the two solutions. (But note that it is not clear whether considering only pairs of trains will be sufficient in general).

It is interesting to observe that these simple subproblems are sufficient to solve the problem illustrated in Figure 3. In fact, the right picture shows the solution obtained by the Lagrangian relaxation with these subproblems, which is already an integral solution.

5.4 Configurations with One Siding

As already observed in [12], enforcing compatible orderings along non-overtaking sections improves the relaxation, but more complex situations are still problematic. Indeed, we consider the case with a single siding between two non-overtaking sections. The best that can be achieved with the approach of Section 5.3 is to enforce compatible configurations on the section left of the siding and on the right of the siding. But the structural influence of the overtaking possibility at the siding is not covered well. For illustration, consider the case where *three* trains run on this track section from left to right. The additional side track at

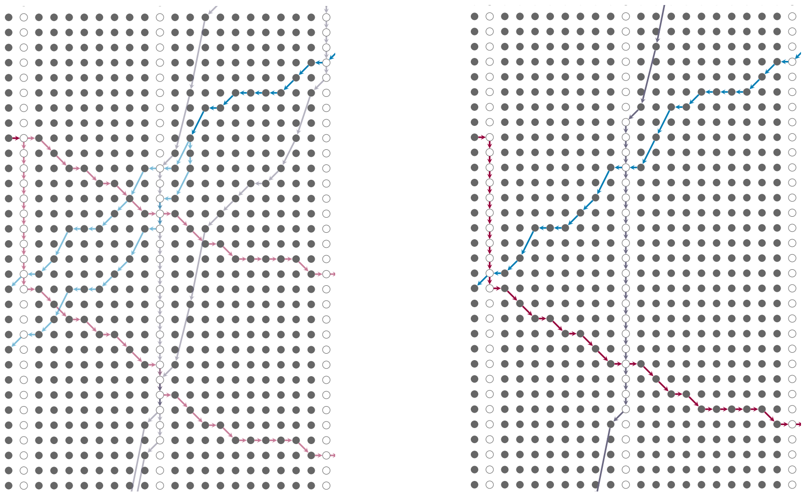

Figure 4 Relaxation of three trains running on two non-overtaking sections with one siding in between. The left picture shows the relaxation with perfect configurations on both non-overtaking sections only. All arcs have a flow of 0.5. One can check that this is not a convex combination of feasible integer solutions. The right picture shows the same situation with an additional capacity-2 configuration for the siding.

Figure 5 The left picture shows the structure of single-arc only subproblems. The right picture shows the case with the new siding configuration, which covers the arcs adjacent to v_s.

the siding allows the order of the trains to be changed (e. g. the first train enters the siding and waits for the second train to pass), but not arbitrarily. For instance, the order of the three trains might not be reversed: if three trains A, B and C enter the section in this order from the left but would leave the section in order C, B, A at the right, then both trains, A and B would have to wait somewhere for C to pass. But this is impossible because there is only one side track. Similar situations occur for trains running in opposite directions. Indeed, Figure 4 shows an example for three trains where only compatibility along the non-overtaking parts is enforced.

In order to guarantee the validity of the change of train orders in the siding, we encapsulate the siding node along with the two adjacent arcs in a single configuration subproblem. Formally, consider a track section $P^I = v_0 a_1 v_1 \ldots a_s v_s a_{s+1} \ldots a_k v_k$ with $a_i \in A^I$, where $c_{v_i} = 1$ for all $i \neq s$ and $c_{v_s} = 2$ and all arcs are single track. Therefore, overtaking can only happen if some train stops and waits at node v_s until some other train passes. Similar as before we split the path in several subpaths each consisting of a single infrastructure arc with one exception: the siding node v_s and its two adjacent arcs are put in a single subpath: $P^I_i = v_{i-1} a_i v_i$, $i = \{1, \ldots, k\} \setminus \{s, s+1\}$, and $P^I_s = v_{s-1} a_s v_s a_{s+1} v_{s+1}$. Figure 5 visualises the structure of the configuration subproblems.

As before overtaking is not allowed on the single-arc paths, so $\pi_i^{\text{in}} = \pi_i^{\text{out}}$ for all $i \in \{1, \ldots, k\} \setminus \{s, s+1\}$. However, this is not true any longer for the two orderings associated with the siding, here π_s^{in} and π_s^{out} may be different, although not all combinations are possible (e. g. the order of three trains must not be reversed). In sum, we need to solve $|R'|!$ subproblems (one for each order of trains) on each single-arc subpath and at most $(|R'|!)^2$ subproblems for the siding subpath.

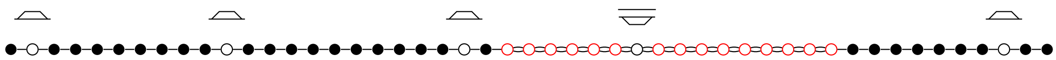

Figure 6 The infrastructure network of the test instance. The black nodes denote single line parts, the red nodes double line parts. There are four sidings (white nodes) within the single line parts and one siding in the double line part, which can only be used by trains coming from the left.

Obviously, the siding subproblems are the most challenging ones and take the longest time, hence we need to keep the subproblems as simple as possible. Hence, we restrict to small subsets of trains with $|R'| = 3$. In this case the analogue intuition is that three trains is the smallest number of trains that cannot change their order arbitrarily, so if we ensure that for each subset of three trains the configurations are valid then it might be sufficient for larger subsets. Indeed, the right picture of Figure 4 shows the solution of the relaxation if the siding configuration subproblem is used, which is integral.

▶ Remark. The trains considered in a configuration do not necessarily need to run on the same track. For instance, if there are at least three trains, each may arrive at node v_s from a different direction in a star-like fashion. Similarly, one of the adjacent edges could be double track whereas the other is single track, so trains running in opposite directions do not share the same physical track. Both cases lead to slightly different configuration subproblems around the node v_s. Because the network considered in our computational tests is a corridor, the first situation cannot happen (there are only two possible directions), but the second situation does appear at the ends of the double track section.

6 Computational Experiments

In this section we present some promising computational results. We consider an instance from the INFORMS RAS Problem Solving Competition 2012 [22] with only 14 trains. The train graph subproblems have been implemented and solved using the DynG callable library [13, 14] and the configuration subproblems are modelled as MIP and solved by Cplex 12.7 [18].

The Lagrangian relaxation is solved with a proximal bundle method based on [16]. All experiments are done on an Intel Xeon E5–2690 with 56 cores @ 2.6 GHz and 256 GB RAM. The evaluation of each subproblem runs in its own thread.

We compare the solutions of the Lagrangian relaxation of three different models:

Free: The basic model (TTP-cfg), configurations on single infrastructure arcs.

Simple: Connected configuration subproblems only on non-overtaking sections (see Section 5.3).

Full: Connected configuration subproblems on non-overtaking sections and sections with one siding (see Section 5.4).

Each connected configuration subproblem contains at most three trains.

First we compare the development of the objective values of the relaxation. Figure 7 shows the (dual) objective value after a certain number of iterations of the bundle method and after some time for the three models.

The first observation is that the objective value of model "free" is much smaller than the objective values of the other models. This is not surprising: because of the weak formulation most trains are hardly slowed down at all (see Section 5). Because the objective function penalises delays from the fastest possible train schedules, this leads to a very small objective value and therefore to a huge gap. The objective values of the two other models are much larger and very similar with "full" having a slightly larger value. It is clear that the objective

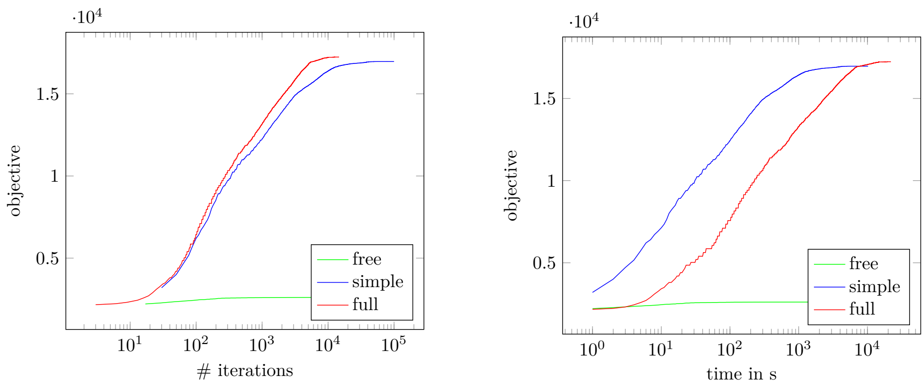

Figure 7 Objective value after a number of iterations (left picture) and after some time (right picture) for three models: classic subproblems without ordering coupling (free), with simple coupling along non-overtaking (simple) and with full coupling at sidings of capacity one (full).

value of “full” is the largest because it is the strongest model. It might, however, be a little surprising that this small difference results in a solution of apparently higher quality (see below). This is an important observation because it indicates that we need to solve the Lagrangian relaxation to a high accuracy in order to gain something from using the stronger model.

The second observation is that the “full” model takes longer (in terms of computation time) to converge to its optimal solution. The reason is that this model contains the most complex configuration subproblem. In fact, in our tests the running time of the overall algorithm was dominated by the solution of a single configuration subproblem for the siding case, which took about 1-2 seconds per iteration. This sounds not that much, but recall that the algorithm needs about 10^3 to 10^4 iterations, so it sums up. Although our implementation takes advantage of the multi-core CPU by solving all subproblems in parallel, each single subproblem is solved on a single core, so the slowest subproblem dominates the running time.

Next we take a closer look at the computed solutions of the relaxation. Figures 8 to 10 show the solutions for the three modes “free”, “simple” and “full”.

The solution of model “free” in Figure 8 shows the expected behaviour: by simply splitting the train schedules into multiple fractional parts, the trains can pass/overtake each other even at non-overtaking sections and are hardly slowed down. The result is that the trains have almost no delay, but the solution is also a very bad starting point for obtaining the real order of the trains in order to derive an integer solution. This is the typical case in classic (time expanded) models for the TTP and one reason why it is hard to find provably good integer solutions.

The solution of the second model “simple” in Figure 9 shows a different picture. This model handles the meet/overtaking decisions on non-overtaking sections correctly. In fact, all decisions have been resolved correctly for the earlier time steps (left picture of Figure 9). The weakness of this model comes apparent if more than 2 trains meet at some siding (see the right picture of Figure 9). In this case the relaxation may still make incorrect decisions resulting in fractional train schedules and unclear orders of trains.

All of these issues have been resolved in the strongest model “full”, see Figure 10. This model contains full configurations around the sidings for each 3 trains. In fact, in this example the solution of the Lagrangian relaxation of the model “full” is *integral*. This cannot be expected in general but demonstrates the strength of the proposed formulation. Even if

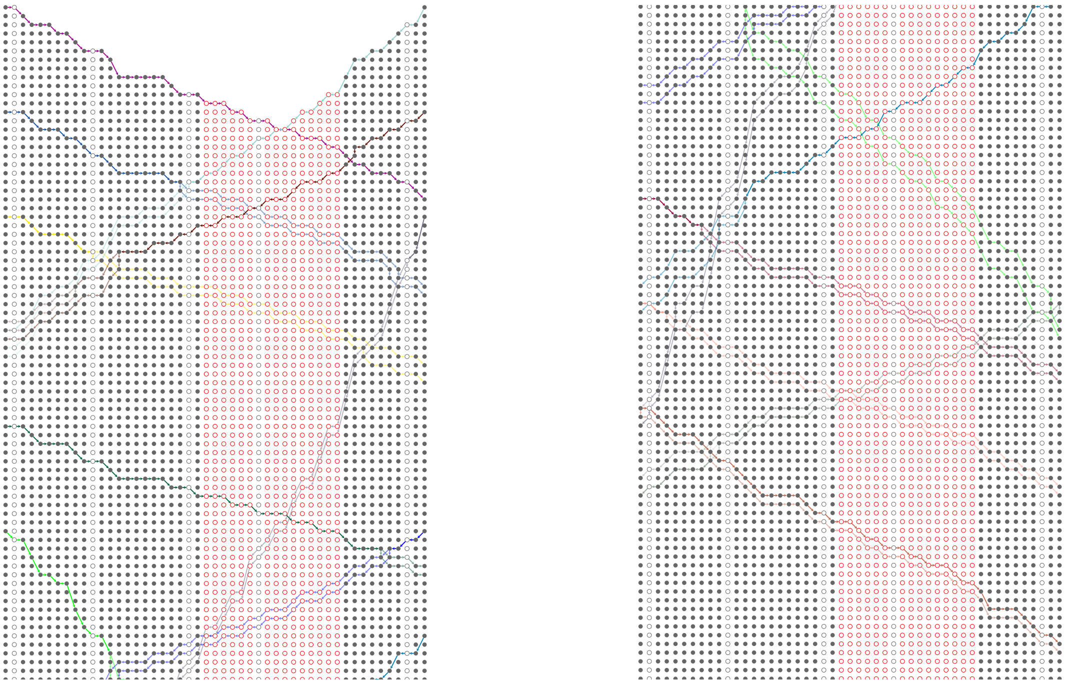

Figure 8 Solution of the relaxation for model *free*. The time runs from top to bottom. By exploiting the weakness of the formulation described in Section 5 many trains meet and pass in non-overtaking sections, splitting the trains in several fractional parts.

the solution might not be integral for larger instances, they certainly provide much more information and much better bounds a subsequent (heuristic) method can use for constructing an integral solution.

7 Conclusions and Future Work

Classic, time expanded models for the TTP are computationally very challenging. One reason for this are the weak relaxations for the standard models, which do not represent combinatorial properties of the problem very well and, hence, do not provide strong bounds. In particular the interplay of headway constraints and node capacities is not handled well.

In this paper we extend the configuration based models proposed in [1]. Instead of modelling feasible configurations on single infrastructure arcs only, we propose a model where configurations of consecutive track segments are handled together. These connected configuration problems are very hard themselves in general, but can be solved reasonably well (using complete enumeration) for small numbers of trains and on specific track segments with only limited overtaking capabilities. The ability to solve these subproblems exactly is then used in a classic Lagrangian relaxation approach. Our computational experiments show that the proposed model and solution approach are able to resolve all overtaking/meet decisions correctly for an instance on a mostly single track corridor with only few sidings, producing even integral solutions in the relaxation. This instance could not be solved to proven optimality using the classic model.

There are several directions for future research. First, for larger instances it is infeasible to add *all* possible configuration subproblems to the model (even for subsets of up to only 3 trains) or to add them manually beforehand. Therefore, a method to automatically detect

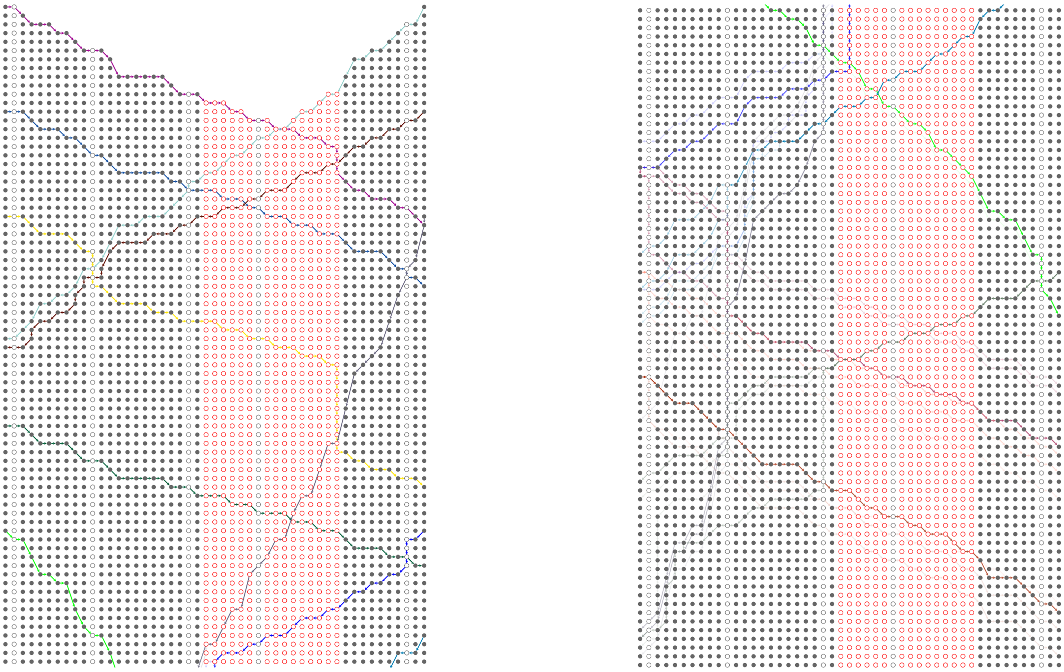

Figure 9 Solution of the relaxation for model *simple*. The time runs from top to bottom. The connected configurations ensure that overtaking/meet decisions for pairs of trains along non-overtaking sections are resolved. However, if more than 2 trains meet at some siding, the fractional solution still contains unclear decisions (right picture).

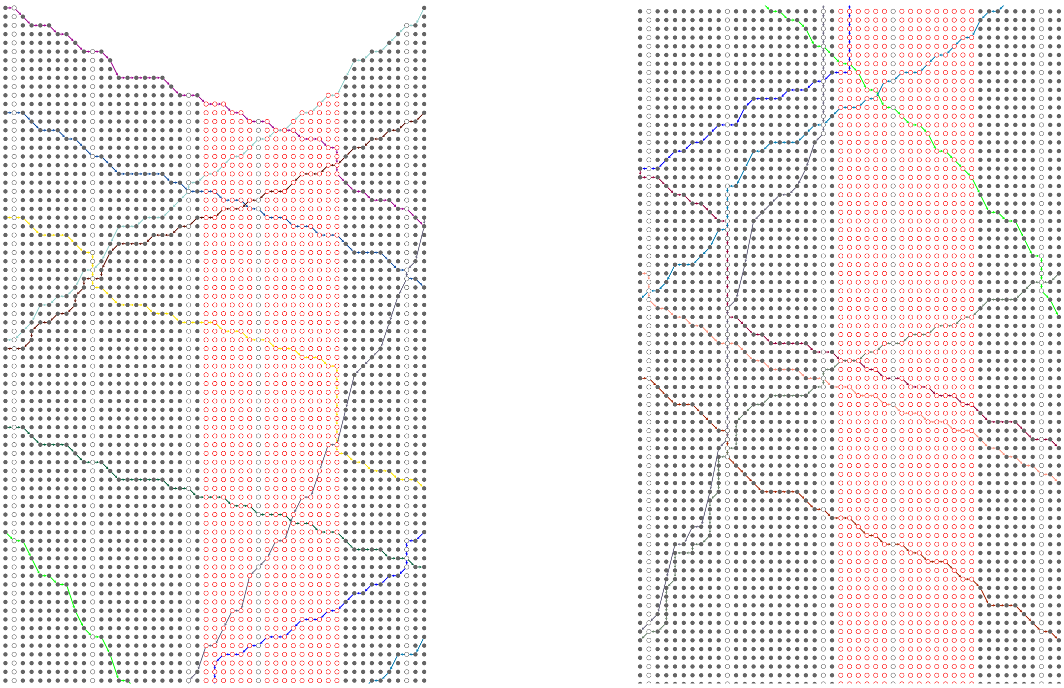

Figure 10 Solution of the relaxation for model *full*. The time runs from top to bottom. The model contains connected configurations on non-overtaking sections and on sections with sidings of capacity two. All decisions have been resolved and (in this example) the solution is even integral.

"missing" configuration subproblems during the solution process and add them to the model on-the-fly while separating the coupling constraints needs to be developed. Secondly, the model presented in this paper handles only cases with no siding or exactly one siding, larger examples are expected to require more complex configuration subproblems.

The tests show that the algorithm has a relatively, but not hopelessly, high running time, which should be improved. The connected configuration subproblems are currently solved using a standard MIP solver, which could be done faster by a specialised combinatorial algorithm. The requirement to solve the subproblems to optimality in each iteration could be relaxed, e. g. by using inexact [10] or asynchronous bundle methods [15]. In fact, because the model presented in this paper consists of many hard but widely independent subproblems, it is expected that the algorithm is very well parallelisable and scales well by increasing the number of parallel processes (more trains or tracks just mean more independent subproblems). An implementation for distributed computation is a logical step.

Besides all of these challenging development paths, a near-time goal is to apply our solution approach to larger, publicly available instances, e. g. from TTPLib [23].

References

1. Ralf Borndörfer and Thomas Schlechte. Models for railway track allocation. In Christian Liebchen, Ravindra K. Ahuja, and Juan A. Mesa, editors, *ATMOS 2007 – 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, volume 7 of *OpenAccess Series in Informatics (OASIcs)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2007. URL: `http://www.dagstuhl.de/dagpub/978-3-939897-04-0`, `doi:10.4230/OASIcs.ATMOS.2007.1170`.
2. Ulf Brännlund, Per Olov Lindberg, Andreas Nou, and Jan-Eric Nilsson. Railway timetabling using Lagrangian relaxation. *Transportation Science*, 32(4):358–369, 1998.
3. Valentina Cacchiani, Alberto Caprara, and Paolo Toth. Non-cyclic train timetabling and comparability graphs. *Operations Research Letters*, 38(3):179–184, 2010. `doi:10.1016/j.orl.2010.01.007`.
4. Valentina Cacchiani, Alberto Caprara, and Paolo Toth. Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological*, 44(2):215–231, 2010.
5. Valentina Cacchiani, Dennis Huisman, Martin Kidd, Leo Kroon, Paolo Toth, Lucas Veelenturf, and Joris Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37, 2014. `doi:10.1016/j.trb.2014.01.009`.
6. Valentina Cacchiani and Paolo Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737, 2012.
7. Gabrio Caimi, Leo G. Kroon, and Christian Liebchen. Models for railway timetable optimization: Applicability and applications in practice. *JRTPM*, 6(4):285–312, 2017. `doi:10.1016/j.jrtpm.2016.11.002`.
8. Alberto Caprara, Matteo Fischetti, and Paolo Toth. Modeling and solving the train timetabling problem. *Operations Research*, 50(5):851–861, 2002.
9. Alberto Caprara, Michele Monaci, Paolo Toth, and Pier Luigi Guida. A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Appl. Math.*, 154(5):738–753, 2006. `doi:10.1016/j.dam.2005.05.026`.
10. Wellington de Oliveira, Claudia Sagastizábal, and Claude Lemaréchal. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Mathematical Programming*, 148(1):241–277, 2014. `doi:10.1007/s10107-014-0809-6`.

11 Frank Fischer. *Dynamic Graph Generation and an Asynchronous Parallel Bundle Method Motivated by Train Timetabling.* PhD thesis, Chemnitz University of Technology, 2013. URL: `http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-118358`.

12 Frank Fischer. Ordering constraints in time expanded networks for train timetabling problems. In Giuseppe F. Italiano and Marie Schmidt, editors, *15th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, ATMOS 2015, September 17, 2015, Patras, Greece*, volume 48 of *OpenAccess Series in Informatics (OASIcs)*, pages 97–110. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. URL: `http://www.dagstuhl.de/dagpub/978-3-939897-99-6`, `doi:10.4230/OASIcs.ATMOS.2015.97`.

13 Frank Fischer. *DynG Dynamic Graph Generation library 0.4.1*, 2017. URL: `http://www.mathematik.uni-kassel.de/~fifr/fossils/cpp-dyng`.

14 Frank Fischer and Christoph Helmberg. Dynamic graph generation for the shortest path problem in time expanded networks. *Mathematical Programming A*, 143(1-2):257–297, 2014. `doi:10.1007/s10107-012-0610-3`.

15 Frank Fischer and Christoph Helmberg. A parallel bundle framework for asynchronous subspace optimization of nonsmooth convex functions. *SIAM Journal on Optimization*, 24(2):795–822, 2014. `doi:10.1137/120865987`.

16 Christoph Helmberg. *ConicBundle 0.3.11.* Fakultät für Mathematik, Technische Universität Chemnitz, 2012. `http://www.tu-chemnitz.de/~helmberg/ConicBundle`.

17 Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms I & II*, volume 305, 306 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 1993.

18 *IBM ILOG CPLEX 12.7, User's Manual for CPLEX, 2016.* URL: `https://www.ibm.com/support/knowledgecenter/SSSA5P_12.7.0`.

19 Leonardo Lamorgese, Carlo Mannino, and Mauro Piacentini. Optimal train dispatching by Benders'-like reformulation. *Transportation Science*, 50(3):910–925, 2016. `doi:10.1287/trsc.2015.0605`.

20 Sascha G. Lukac. Holes, antiholes and maximal cliques in a railway model for a single track. Technical Report ZIB Report 04-18, Zuse-Institut Berlin, Takustr. 7, 14195 Berlin, 2004. URL: `http://www.zib.de/PaperWeb/abstracts/ZR-04-18`.

21 Alessandro Mascis and Dario Pacciarelli. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143(3):498–517, 2002. `doi:10.1016/S0377-2217(01)00338-1`.

22 RAS Problem Solving Competition 2012, 2012. URL: `https://www.informs.org/Community/RAS/Problem-Solving-Competition/2012-RAS-Problem-Solving-Competition`.

23 TTPlib-Homepage, 2008. http://ttplib.zib.de. URL: `http://ttplib.zib.de`.