Real Time Railway Traffic Management Modeling Track-Circuits

Paola Pellegrini, Grégory Marlière, and Joaquin Rodriguez

Ifsttar – ESTAS
Univ. Lille Nord de France
rue Élisée Reclus 20, 59666 Villeneuve d'Ascq, Lille, France
`paola.pellegrini@ifsttar.fr`, `gregory.marliere@ifsttar.fr`,
`joaquin.rodriguez@ifsttar.fr`

Abstract

The real time railway traffic management seeks for the train routing and scheduling that minimize delays after an unexpected event perturbs the operations. In this paper, we propose a mixed-integer linear programming formulation for tackling this problem, modeling the infrastructure in terms of track-circuits, which are the basic components for train detection. This formulation considers all possible alternatives for train rerouting in the infrastructure and all rescheduling alternatives for trains along these routes. To the best of our knowledge, we present the first formulation that solves this problem to optimality. We tested the proposed formulation on real perturbation instances representing traffic in a control area including the Lille Flandres station (France), achieving very good performance in terms of computation time.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases real time railway traffic management, mixed-integer linear programming, track-circuit, complex junction

Digital Object Identifier 0.4230/OASIcs.ATMOS.2012.23

1 Introduction

Railway infrastructure has a limited physical capacity that is often insufficient to smoothly accommodate traffic when unexpected events perturb operations. This insufficiency appears in terms of train *conflicts*: multiple trains concurrently claim a portion of track. In case of conflicts, trains must be delayed for sequencing their use of the critical portion of track. *Junctions* are the physical locations on which conflicts are most likely to occur. In a junction, different lines cross and, often, multiple *routes* can be used for joining an origin to a destination. Considering the railway network from a macroscopic point of view, junctions represent nodes and lines are links among these nodes [13]. Terminal stations are junctions where trains may stop for loading and unloading purposes and where the configuration of both the rolling stock and the crew may be modified.

Traffic on the railway network is managed by *dispatchers*. They are in charge of smoothing operations in their *control areas*. If a control area includes a complex junction, the dispatcher task may become very challenging. Currently, few automatic tools are available for rerouting or rescheduling trains in junctions in real time. The available tools, as for example the ARI system used in the Netherlands, may just reserve routes to trains on the basis of the timetable scheduling and on arrival time forecasts. Despite the undeniable aid of these tools, dispatchers must often take decisions autonomously [4].

Several authors have proposed optimization algorithms for tackling the problem faced by dispatchers. We will refer to the formal problem tackled as the real time railway traffic

12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'12).
Editors: Daniel Delling, Leo Liberti; pp. 23–34
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

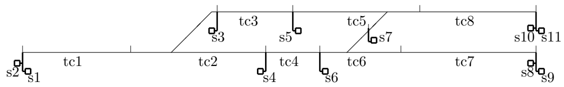

Figure 1 Example of the infrastructure present in a control area.

management problem (rtRTMP). In the literature, different variants of the rtRTMP have been tackled. The first papers that appeared did not considered rerouting possibilities [5, 8], proposing either optimal or heuristic solutions for the rescheduling problem. In the following, other algorithms introduced the possibility of rerouting trains [4, 3, 11, 14, 16, 17], finding heuristic solutions to the rtRTMP. All these algorithms have the characteristic of neglecting speed variation dynamics: they are fixed-speed algorithms. The main reason for this neglect is the fact that the consideration of speed variation dynamics (in variable-speed algorithms) is computationally extremely costly and, then, hardly possible in real time. To the best of our knowledge, two variable-speed algorithms have been proposed in the literature: for being able to take into account speed variation dynamics they either neglect [7] or strongly limit [10] the possibility or rerouting trains. A further possibility that has been explored in the literature is forbidding the imposition of delays within the control area considered: if trains are rescheduled simply by imposing a later entrance time in the control area, then there are no speed variations to be decided, and hence to be accounted for in the optimization [1, 2]. Yet, the drawback of this imposition is the lack of consideration of any constraint outside the control area: this may cause severe coordination problems among control areas.

In this paper, we propose a fixed-speed algorithm, focusing our attention on the potential of rerouting: in our model, we consider all the possible routes that physically exist in the control area. Moreover, we detail the modeling of the control area itself up to the consideration of track-circuits.

A control area, in fact, is composed by portion of tracks on which the presence of a train is automatically detected by an electric mechanism. These portions are called track-circuits. They are typically grouped into block sections started and ended by a light signal. The aspect of the light signal imposes the behavior to be held by the driver entering the block section: proceeding at the scheduled speed (green aspect), braking for being able to stop by the following signal (yellow aspect), or stop (red aspect). Different signaling systems exist, typically with different number of aspects: three being the most common configuration, further aspects may separate the green and the red one, with a consequently larger number of block sections for braking. In general terms, if the signaling system has n possible aspects, then $n-2$ block sections are available for braking. Figure 1 depicts an example of the infrastructure characterizing a control area. Track circuits are named tc and signals are named s, both indexed with a progressive number. Signals concern the availability of block sections in a precise direction: for example, signal s1 concerns block section s1-s5 including tc1, tc2 and tc3, in this order, and block section s1-s6 including tc1, tc2 and tc4, in this order. When a train enters a track-circuit, all the following ones belonging to the same block section are reserved for the train itself.

The algorithms proposed in the literature consider alternatively track-circuits, block sections or track sections including a number of block sections as smallest decomposition of the infrastructure. Considering track-circuits allows the full exploitation of the capacity of the control area: in the example, if a train is known to be in track-circuit tc3, going from tc1 to tc8, then block section s1-s6 is available only if the model considers track-circuits. Otherwise, it is not possible to distinguish the presence of a train on tc1, tc2 or tc3, and hence both s1-s6 and s1-s5 are unavailable as long as the train has not entered the following block section.

A further issue that emerges is the realism of the representation: often block sections for routes in opposite directions do not coincide. For example, consider the routes tc1 to tc8 and tc8 to tc1. In the former, tc1, tc2 and tc3 belong to the first block section and tc5 and tc8 to the second one. In the latter, tc8, tc5 and tc3 belong to the first block section and tc2 and tc1 to the second one. When representing the control area considering block sections, it is not clear where track-circuit tc3 should be positioned. Wherever it is positioned, the model will not represent the real infrastructure.

Our formulation deals with a track-circuit based model, thus allowing the full exploitation of capacity and the realistic representation of the infrastructure. Of course, the number of variables to be included in the model increases very fast with the size of the control area. Yet, in the experimental analysis we propose, we show that our formulation can deal with instances representing rather large control areas. In particular, it solves in few minutes instances obtained by perturbing real instances representing the control area including the Lille Flandre station, in France.

The rest of the paper is organized as follows. Sections 2 and 3 depict the main characteristics of the rtRTMP and the formulation that we are proposing in this paper, respectively. Section 4 presents the experimental setup and the instances tackled and Section 5 shows the results of the analysis. Finally, Section 6 concludes the paper.

2 The real time railway traffic management problem

When an unexpected event occurs, trains suffer a non-negative delay at their entrance in the control area. This delay is typically named primary delay and it may cause the emergence of conflicts within the control area itself. The additional delay due to these conflicts is named secondary delay. According to the literature [5], the objective of the rtRTMP is the minimization of the maximum secondary delay assigned to trains. Multiple sets of constraints characterize this problem.

First of all, **time concerning constraints**: a train cannot be scheduled earlier than its entry time (if starting within the control area, the planned departure time is considered as entry time) and it must occupy each track-circuit along one route for a certain amount of time. In variable-speed models, this time depends on traffic conditions. In fixed-speed models, it is computed a priori as the *running time* in absence of conflicts. The time in which the train occupies two consecutive track-circuits is named *clearing time*: its rear is still on the current track-circuit and its front has already entered the following one. Before a train enters a block section, some time must be allowed for route formation and for taking into account the signal visibility distance [18]. In the following we will refer to the sum of these times simply as *formation time*. After a train exits the block section, some time must elapse before the block section become available for another train. In this time, the route is released (*release time*) [18]. Finally, if the control area includes a station and trains with passenger transfers (*trains in connection*) are scheduled, then their arrival and departure time must be coherent.

Second, some **constraints for managing delays** may be imposed. Three cases are possible: no constraints, delay allowed at any signal and delay allowed only out of the control area. In absence of constraint, delay may be assigned anywhere in the control area: the underlying hypothesis is that the dispatcher can stop the train in any track-circuit along the route. The case of delay allowed at any signal represents the fact that, in reality, trains stop in front of signals. If delay can be assigned only out of the control area, no difference exists between fixed and variable-speed models, but complex coordination issues between control areas may emerge, as mentioned in the introduction.

Third, **constraints due to the change of rolling stock configuration** may have to be imposed. In particular, the arrival and departure time of trains resulting from the turn-around, join or split of one another must be coherent.

Fourth, **capacity constraints** require that at most one train occupies a block section at a time. All track-circuits belonging to a block section must be reserved for a train before it enters it. When designing timetables, *blocking times* are considered for having a separation between consecutive trains that allows them to always encounter green aspect signals [12]. In particular, these blocking times include the approach time that is often set equal to the total running time of all track-circuits following the first restricted signal (aspect different from the green one). When applying this concept (*blocking time theory*) to the rtRTMP, this translates into a set of constraints imposing that the reservation of a track-circuit starts as soon as the train enters in the first track-circuit of the preceding block section.

3 Mixed-integer linear programming formulation

In the formulation proposed, for coping with the fact that a track-circuit may require different running times depending on the route on which it is used (for example, if a non-negligible slope characterizes the terrain, then the running time may be much different for trains running in opposite directions) we consider a set of nominal track-circuits: we duplicate each real track-circuit as many times as the number of different routes using it. Each nominal track-circuit belongs to a single route and has a single running time. In the following, we will distinguish the reference to either nominal or real track-circuits whenever necessary.

Moreover, we introduce two dummy track-circuits: tc_0 and tc_∞. They represent the entry and the exit locations of the control area, respectively. The former precedes all actual entry track-circuits in the control area and the latter follows all actual exit ones. Their running time is null.

In the following, we describe the objective function and the constraints defining the formulation through the following notation and variables. The constraints presentation follows the problem description in Section 2. For sake of brevity, we do not explicitly report integrality and non-negativity constraints.

T, R	set of trains and routes, respectively,
RTC, TC	set of real and both nominal and dummy (from here on, *nominal*) track-circuits, respectively,
$S \subseteq T$	trains representing shunting movements,
RTC_t, TC_t	set of real and nominal track-circuits that can be used by train t, respectively,
$PL \subset RTC$	set of real track-circuits corresponding to platforms,
$R_t \subseteq R$	set of routes that can be used by train t,
RTC^r	set of real track-circuits composing route r,
rtc^{tc}	real track-circuit corresponding to the nominal one tc ($tc \notin \{tc_0, tc_\infty\}$),
r^{tc}, bs_{tc}	route and block section including nominal track-circuit tc ($tc \notin \{tc_0, tc_\infty\}$), respectively,
p_{tc}, s_{tc}	preceding ($tc \neq tc_0$) and subsequent ($tc \neq tc_\infty$) nominal track-circuit of tc, respectively,

$ebs(tc)$	indicator function ($tc \notin \{tc_0, tc_\infty\}$): 1 if nominal track-circuit tc belongs to an extreme (either the first or the last) block section on its route, 0 otherwise,
ref_{tc}	reference nominal track-circuit for the reservation of tc ($tc \notin \{tc_0, tc_\infty\}$). In a two-aspect system, it is the first nominal track-circuit of bs_{tc}. In a three-aspect system, it is the first nominal track-circuit of the block section preceding bs_{tc},
run_{tc}, cl_{tc}	running and clearing time of nominal track-circuit tc, respectively,
$form, rel$	formation and release time, respectively
$init_t, sched_t$	train t foreseen entry time and scheduled exit time of train t, respectively. If t is subject to primary delay, $sched_t$ is equal to the scheduled exit time plus the primary delay itself,
$I(t, t')$	indicator function: 1 if train t' results from the turn-around, join or split of train t, 0 otherwise,
$RoSt(t, t')$	indicator function: 1 if trains t and t' use the same rolling stock, 0 otherwise,
$C(t, t')$	indicator function: 1 if trains t and t' are in connection, 0 otherwise,
ms, ms_c	minimum separation between the arrival of a train and the departure of another train using the same rolling stock, or of a train in connection, respectively,
M	large constant.

We define both continuous and binary decision variables. First of all, we define

$$D = \text{maximum delay assigned to any train.}$$

Moreover, we define **continuous variables** for: all pairs of train $t \in T$ and nominal track-circuit $tc \in TC_t$:

$$e_{t,tc} = \text{time in which } t \text{ enters } tc, \qquad d_{t,tc} = \text{delay assigned to } t \text{ in } tc \text{ (defined if } bs_{tc} \neq bs_{s_{tc}});$$

all pairs of train $t \in T$ and real track-circuit $rtc \in RTC_t$:

$$sRes_{t,rtc} = \text{time in which } rtc \text{ starts being reserved for } t,$$
$$eRes_{t,rtc} = \text{time in which } rtc \text{ ends being reserved for } t.$$

We define **binary variables** for: all pairs of train $t \in T$ and route $r \in R_t$:

$$x_{t,r} = \begin{cases} 1 & \text{if } t \text{ uses } r, \\ 0 & \text{otherwise;} \end{cases}$$

all triplets of train $t, t' \in T$ and real track-circuit $rtc \in RTC_t \cap RTC_{t'}$:

$$y_{t,t',rtc} = \begin{cases} 1 & \text{if } t \text{ uses } rtc \text{ before } t' \ (t \prec t'), \\ 0 & \text{otherwise } (t \succ t'). \end{cases}$$

Figure 2 shows the role of these variables in a portion of the example depicted in Figure 1, corresponding to train t on block section s1-s5 along route s1-s5, s5-s11 (named r1). Nominal track-circuits along this route are named r1_tc1, for example considering real track-circuit tc1. We depict track-circuit occupation as a rectangle with solid borders and reservation as a rectangle with dashed borders: the horizontal dimension represents time. The reservation of a track-circuit starts *form* time units before the physical occupation of the first track-circuit in the block section, and it ends *rel* time units after the end of the occupation of the track-circuit itself. Each track-circuit is physically occupied for a running time *run* plus a clearing time *cl*: they both depend on the track-circuit and on the route on which it is used.

3.1 Objective function and constraints

As mentioned in Section 2, the objective of the rtRTMP is the minimization of the maximum secondary delay imposed to a train: the objective function is

$$\min D.$$

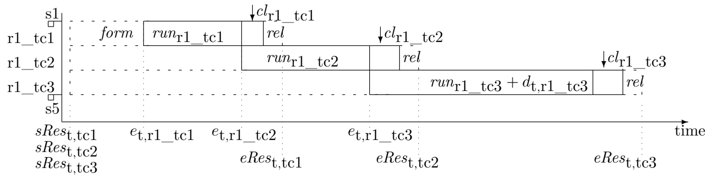

Figure 2 Graphical representation of data and variables. Subset of nominal track-circuits shown in Figure 1, belonging to block section s1-s5.

Time concerning constraints

$$e_{t,tc} \geq init_t \; x_{t,r^{tc}} \quad \forall t \in T,\; tc \in TC_t, \tag{1}$$

$$e_{t,tc} \leq M x_{t,r^{tc}} \quad \forall t \in T,\; tc \in TC_t, \tag{2}$$

$$e_{t,tc} \geq e_{t,p_{tc}} + run_{p_{tc}} x_{t,r^{tc}} \quad \forall t \in T,\; tc \in TC_t \setminus \{tc_0, tc_\infty\}, \tag{3}$$

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T, \tag{4}$$

$$\sum_{\substack{tc \in TC_{t'}:\\ p_{tc} = tc_0}} e_{t',tc} \geq \sum_{\substack{tc \in TC_t:\\ s_{tc} = tc_\infty}} e_{t,tc} + (ms_c + run_{tc}) x_{t,r^{tc}} \quad \forall t, t' \in T : C(t,t') = 1, \tag{5}$$

$$D \geq e_t - sched_t \quad \forall t \in T \setminus S. \tag{6}$$

Constraints (1) state that trains cannot be scheduled earlier than their entry time in the control area. Constraints (2) impose that the entry time in a nominal track-circuit is set to 0 if the route to which it belongs is not used. Recall that each nominal track-circuit belongs to one and only one route. Constraints (3) state that a train cannot enter tc if it has not spent in its preceding track-circuit at least its running time, if they are used. Constraints (4) ensure that exactly one route is used by each train. If the control area includes a station and trains in connection are scheduled, then we must impose Constraints (5). They state that a minimum separation of duration ms_c must be ensured between trains arrivals and departures. The spatial coherence is ensured by the routes available for the trains: any route available for the arriving train terminates at a platform and any route available for the departing one starts from a platform. Constraints (6) impose the coherence of variable D. Here we do not consider delay assigned to shunting movements in the objective function, hence we impose these constraints for all trains in $T \setminus S$. For taking into account also shunting movements, is suffices to impose them for all trains in T.

Constraints for managing delay

$$d_{t,tc} = e_{t,s_{tc}} - e_{t,tc} - run_{tc} x_{t,r^{tc}} \quad \forall t \in T,\; tc \in TC_t : bs_{tc} \neq bs_{s_{tc}} \tag{7}$$

$$e_{t,tc} = init_t \; x_{t,r^{tc}} \quad \forall t \in T \setminus S : tc \in TC_t : p_{tc} = tc_0,\; rtc^{tc} \notin PL. \tag{8}$$

For each nominal track-circuit tc that can be used by train t and that closes its block section, delay variable $d_{t,tc}$ assumes value equal to the moment in which train t enters the nominal track-circuit that follows tc, minus the moment in which it enters tc itself, minus the running time run_{tc}: Constraints (7) ensure this relation. Constraints (8) impose that trains are not delayed before entering the control area, unless they depart from a platform or they represent shunting movements: train t enters the first nominal track-circuit exactly at time $init_t$ along the route selected.

Constraints due to the change of rolling stock configuration

$$\sum_{\substack{tc \in TC_t: \\ p_{tc} = tc_0}} e_{t,tc} \geq \sum_{\substack{tc \in TC_{t'}: \\ s_{tc} = tc_\infty}} e_{t',tc} + (ms + run_{tc}) x_{t',r^{tc}} \quad \forall t,t' \in T : I(t',t) = 1, \tag{9}$$

$$\sum_{\substack{tc \in TC_t: \\ p_{tc} = tc_0}} sRes_{t,rtc^{tc}} \leq \sum_{\substack{tc \in TC_{t'}: \\ s_{tc} = tc_\infty}} eRes_{t',rtc^{tc}} \quad \forall t,t' \in T : I(t',t) = 1, \tag{10}$$

$$\sum_{r \in R_t : rtc \in PL \cap RTC_t} x_{t,r} = \sum_{r \in R_{t'} : rtc \in PL \cap RTC_{t'}} x_{t',r} \quad \forall t,t' \in T : I(t',t) = 1, rtc \in PL. \tag{11}$$

Similarly to Constraints (5), Constraints (9) state that a minimum separation of duration ms must be ensured between t''s arrival and t's departure, if t results form t''s turn-around, join or split. Constraints (10) ensure that the real track-circuit where the turn-around, join or split takes place is reserved by t' until it arrives at the platform, plus the release time, and then it is immediately reserved by t. We must impose an inequality for allowing joins: the reservation of the resulting train starts with the ending of the reservation of the first train arriving. We manage the capacity issues arising with two trains reserving concurrently a real track-circuit as described in the next section. Besides the train temporal coherence, we must ensure local coherence: trains using the same rolling stock must use routes including the same platform. Constraints (11) guarantee this local coherence. Of course, if the routes available for the two trains share only one platform, i.e., if the dispatcher is not allowed to impose platform changes, these constraints will be trivially met.

Capacity constraints

$$sRes_{t,rtc} = \sum_{tc \in TC_t : rtc^{tc} = rtc} e_{t,ref_{tc}} - form\, x_{t,r^{tc}} \; \forall t \in T,\; rtc \in RTC_t, \tag{12}$$

$$eRes_{t,rtc} = \sum_{tc \in TC_t : rtc^{tc} = rtc} e_{t,s_{tc}} + (cl_{tc} + rel)\, x_{t,r^{tc}} \; \forall t \in T,\; rtc \in RTC_t, \tag{13}$$

$$y_{t,t',rtc} + y_{t',t,rtc} = 1 \; \forall t,t' \in T, rtc \in RTC_t \cap RTC_{t'}, \tag{14}$$

$$\sum_{\substack{tc \in TC_t : rtc^{tc} = rtc, \\ RoSt(t,t') ebs(tc) = 0}} eRes_{t,rtc} - M(1 - y_{t,t',rtc}) \leq \tag{15}$$

$$\sum_{\substack{tc \in TC_t : rtc^{tc} = rtc, \\ RoSt(t,t') ebs(tc) = 0}} sRes_{t',rtc} \; \forall t,t' \in T : rtc \in RTC_t \cap RTC_{t'}$$

$$\sum_{\substack{tc \in TC_t : rtc^{tc} = rtc, \\ RoSt(t,t') ebs(tc) = 0}} eRes_{t',rtc} - M y_{t,t',rtc} \leq \tag{16}$$

$$\sum_{\substack{tc \in TC_t : rtc^{tc} = rtc, \\ RoSt(t,t') ebs(tc) = 0}} sRes_{t,rtc} \; \forall t,t' \in T : rtc \in RTC_t \cap RTC_{t'}.$$

Constraints (12) state that a train's reservation of a real track-circuit starts as soon as the train enters the nominal track-circuit ref_{tc} minus the route formation time. For Constraints (13), the reservation ends as soon as the train has entered the subsequent track-circuit, plus the sum of clearing and release time. Constraints (14) to (16) are disjunctive constraints imposing that real track-circuit reservations do not overlap. Hence, at most

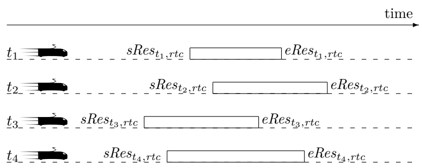

Figure 3 Overlapping of reservation times forbidden by Constraints (14) to (16), using t_1 as reference train. t_1 and t_2: for Constraints (15), $t_1 \prec t_2$ and $y_{t_1,t_2,rtc} = 1 \Rightarrow eRes_{t_1,rtc} \leq sRes_{t_2,rtc}$. t_1 and t_3: for Constraints (16), $t_3 \prec t_1$ and $y_{t_3,t_1,rtc} = 1 \Rightarrow eRes_{t_3,rtc} \leq sRes_{t_1,rtc}$. t_1 and t_4: for Constraints (14) either $t_4 \prec t_1 \Rightarrow y_{t_1,t_4,rtc} = 1$, $y_{t_4,t_1,rtc} = 0$ or $t_4 \prec t_1 \Rightarrow y_{t_1,t_4,rtc} = 0$, $y_{t_4,t_1,rtc} = 1$.

one train reserves a track-circuit at any time and capacity constraints are respected. Constraints (15) and (16) ensure that, if $t \prec t'$ on rtc, then t's reservation ends before the reservation of train t' starts. Instead, if $t' \prec t$ on rtc, then t''s reservation must end before t's reservation can start. Remark that, for ensuring the validity of the constraints, M must be greater than or equal to the maximum time distance between the begin and the end of two trains' reservations of the same track-circuit. If two trains use the same rolling stock, the constraints do not apply to track-circuits belonging to extreme block sections ($RoSt(t,t')ebs(tc) = 1$). Thanks to Constraints (3) and (9), which ensure the time coherence for each train route and for each pair of trains, respectively, any solution imposing a reservation overlap on a real track-circuit, other than the one where the rolling stock configuration change takes place, results infeasible. The fact that a train entering a track-circuit has still an open reservation of the preceding one (for both the clearing and the release time) ensures the feasibility of routes assigned to trains going in opposite directions. Figure 3 shows through three examples how these constraints ensure the feasibility of solutions.

4 Experimental setup

In the experimental analysis, we test our formulation on perturbations of real instances representing traffic in the control area including the main station of Lille in the North of France, i.e., the Lille-Flandres station. In particular, we considered a one-day timetable including 589 trains. Figure 4 shows the temporal distribution of train scheduled entry times in the control area. We do not have any information on connections, and hence we do not consider Constraints (5) presented in Section 3. On the other hand, being the Lille-Flandres station a terminal one, all rolling stocks are used for both an arriving and a departing train, but for what concerns the first trains departing in the morning (which arrived the day before to the platform) and the last ones arriving at night (which will leave the platform the day after): for almost any train t (97.11% of the total) a t' exists such that $RoSt(t,t') = 1$. Besides 259 turn-arounds, the timetable contains 8 joins and 10 splits.

Figure 5 depicts the infrastructure of the control area: the station is linked to seven regional, national and international lines and it has 17 platforms. All routes either depart or arrive at the station: either their initial or their final real track-circuit is a platform. A total of 2409 routes exist and they are composed by 299 real track-circuits. The consequent number of nominal track-circuits is 58748. The routes include 9 to 35 track-circuits (mean = 24), 2 to 13 block sections (mean = 5), and they have a total running time of 2 to 12 minutes (mean = 6) and a total length of 950 to 11500 meters (mean = 4331). More than 85% of real track-circuits belong to non-coincident block sections in the two directions. Hence, if

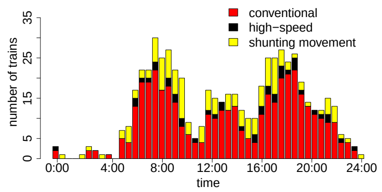

Figure 4 Original timetable: number of trains entering the control area within consecutive half-an-hour time intervals.

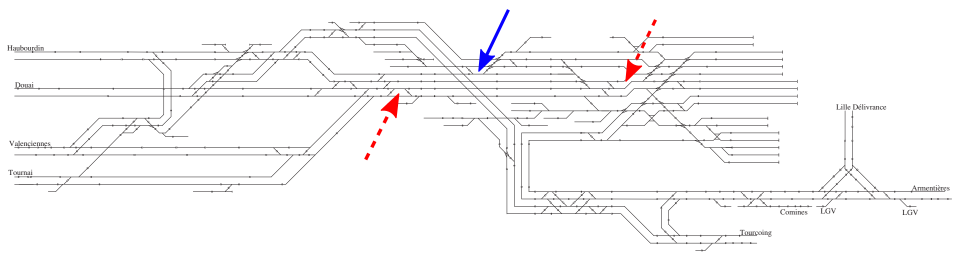

Figure 5 Infrastructure of the control area including the Lille-Flandres station. The blue solid arrow indicates the unavailable track-circuit in the partially disrupted scenario. The red dashed arrows indicate the further track-circuits unavailable in the severely disrupted scenario.

we did not model track-circuits we could not realistically represent this infrastructure.

Starting from the original timetable, we imposed a delay to 20% of trains that do not represent shunting movements: we randomly selected the trains to be delayed and we randomly drew their delay in the interval between 5 to 15 minutes [10]. Both these random selections are based on uniform probability distributions. We replicated the randomly assignment of train delay three times, obtaining three different perturbed timetables. The tackled instances include, for each timetable, all trains arriving in ten half-an-hour intervals along the day: from 7:30 to 9:30 and 16:00 to 19:00, representing the two peak times of the day. Hence, we solve a total of 30 instances with a mean number of train equal to 25.43. In this analysis, we do not consider the existing relation between instances representing consecutive time intervals. Hence, a further procedure (e.g., the one proposed by D'Ariano and Pranzo [6]) shall be used for ensuring global consistency in the daily operations.

We tested our model on the perturbed instances, considering three different scenarios concerning the infrastructure: **fully functioning**, i.e., all existing routes are operational; **partially disrupted**, i.e., one track-circuit is unavailable (indicated with a solid blue arrow in Figure 5) and hence only 67.66% of routes are operational; **severely disrupted**, i.e., three track-circuits are unavailable (indicated with either solid blue or dashed red arrows in Figure 5) and hence only 40.51% of routes are operational. We selected these percentages following the literature [3], and the track-circuit according to the routes they belong to. In these experiments, we consider the platform assigned to trains as non modifiable. Moreover, we use a two-aspect signaling system, which often allows better quality solutions than systems with higher number of aspects, when adopting blocking time theory in a fixed-speed algorithm and assessing solution quality in simulation [15].

We implemented our formulation using the IBM ILOG CPLEX Concert Technology for C++ (IBM ILOG CPLEX version 12) [9] and we ran the experiments on an Intel(R) Xeon(R) quad core 2.93GHz processors with 6 GB RAM, under Linux Ubuntu distribution version

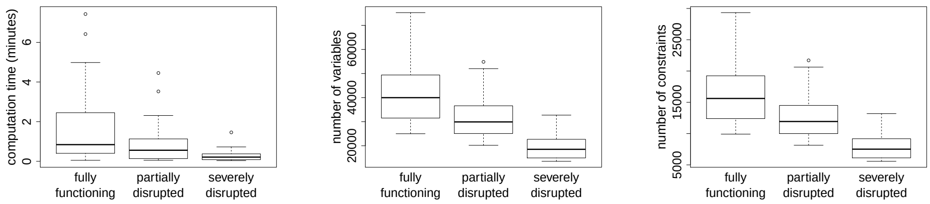

Figure 6 Distribution of the computation time (left), number of variables (center) and constraints (right) in the TC formulation in the three scenarios.

10.04. We selected IBM ILOG CPLEX parameters after a short preliminary analysis on different instances representing the Lille Flandre control area. In particular, we set: BrDir = 0, DiveType = 3, CutsFactor = 1, Probe = 1. For the explanation of the role of these parameters in the solution process we refer the reader to the IBM ILOG CPLEX parameter reference manual [9]. Moreover, we started the solution process by providing an initial solution found by imposing the use of the originally planned routes. The large constant M is set to 86400, i.e., the length of the time horizon: in our instances this quantity is always greater than the time distance between two trains' reservation of a track-circuit, which is necessary for ensuring the validity of Constraints (15) and (16).

We analyze the results achieved by our formulation with respect to the results achieved when considering block sections as smallest decomposition of the control area. By adding a set of constraints imposing that the reservation of all track-circuits belonging to the same block section terminates concurrently, the formulation presented in Section 3 allows modeling the block section decomposition. Differently from the algorithms proposed in the literature, by anyway including track-circuits in the model, we realistically represent the infrastructure when block sections to be used in opposite directions do not coincide. In the following, we will refer to the formulation modeling track-circuits and block sections as the TC formulation and BS formulation, respectively.

5 Computational results

The TC formulation proposed in this paper was able to solve in few minutes all the instances tackled under the three scenarios, with one exception in which the proof of optimality required one hour and twenty minutes (the optimal solution was found after 0.3 minutes).

The left plot in Figure 6 shows the boxplot of the distribution of the computation time needed for finding the optimal solution and proving its optimality (we restricted the margin of the plot excluding the just mentioned outlier for ease of visualization). Here and in the following, computation time is computed in minutes of CPU. Each box represents the observations corresponding to one of the three scenarios studied. The horizontal line within the boxes represent the median of the distributions, while the extremes of the boxes represent the first and third quartiles, respectively; the whiskers show the smallest and the largest non-outliers in the data-set and dots correspond to the outliers. In the great majority of the cases the computation time is **lower than three minutes**, which is typically an accepted time for tackling the rtRTMP in reality [14] (in some cases an even higher time of 4.5 minutes may be accepted [10]). The center and left plots of Figure 6 depict the distribution of the number of variables and constraints in the TC-formulation. The computation time increases as a function of the number of variables and constrains, that in turn vary as a function of the number of available routes for trains and hence of the scenario considered.

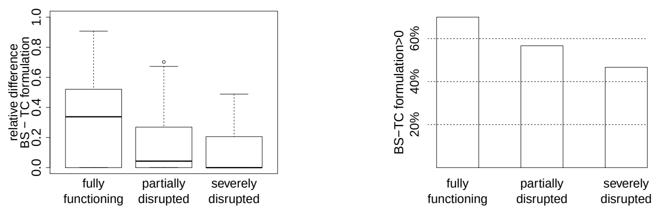

Figure 7 Improvement allowed by the TC formulation over the BS one: distribution of the relative differences (left) and percentage of instances with strictly positive improvement (right).

The difference between the optimal solution values of the BS formulation and of the TC formulation is always non-negative: the feasible region in the BS formulation is a subset of the feasible region of the TC formulation. In all the scenarios, the improvement brought by the TC formulation is statistically significant according to the Wilcoxon rank-sum test with a confidence level of 0.95. The left plot of Figure 7 shows the boxplots of the distribution of the relative difference between the results of the two formulations. As expected, the higher the number of routes available, the larger the improvement allowed by the consideration of track-circuits. In fact, if two trains need to follow each other for a long portion of their route, the second one will not be able to reserve a block section until the first one has exit it, both in the TC and in the BS formulations. Even if in some cases the two formulations return the same solutions, the more efficient use of the infrastructure allowed by the TC formulation allows the reduction of the maximum secondary delay assigned to trains in 58% of the instances. The right plot of Figure 7 depicts the percentage of instances in which the improvement is strictly positive for each scenario.

6 Conclusions

In this paper, we proposed a mixed-integer linear programming formulation for tackling the rtRTMP. It allows splitting routes into track-circuits, i.e., it allows the fine realistic representation of the control area. Moreover, it selects among all possible routes that can be practically exploited and it considers all possible train orderings.

We applied this formulation to instances obtained by perturbing the real timetable of a week day in the control area including the Lille Flandres station, in France, and we considered multiple scenarios in terms of functionality of the infrastructure. The results show that the formulation modeling track-circuits outperforms the more commonly used formulation modeling block sections: the difference between the optimal solution values is statistically significant.

The computation time for solving the instances considered is in line with what is required to a real time algorithm in reality, even after a rough parameter tuning of the exact solver. In future research we devote further effort to boost the performance of our formulation, which anyway already achieves very positive results. We will boost the performance by both fine-tuning parameters and introducing efficient valid inequalities.

Furthermore, we will insert it in a sliding window framework, also known as traffic management system [11] or closed-loop optimization [2]. In this framework, the rtRTMP is solved periodically, considering trains expected to be in the control area during a short time window; at each solution of the rtRTMP, the set of trains to be considered is updated thanks to revised forecasts. Periodical applications of the optimization algorithm allow to solve successive instances and to cover any long time horizon.

References

1 G. Caimi, F. Chudak, M. Fuchsberger, M. Laumanns, and R. Zenklusen. A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling. *Transportation Science*, 45(2):212–227, 2011.

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

3 F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B*, 44:175–192, 2010.

4 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.

5 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.

6 A. D'Ariano and M. Pranzo. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9:63–84, 2009.

7 A. D'Ariano, M. Pranzo, and I.A. Hansen. Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222, 2007.

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

9 IBM Corporation. Ibm ilog cplex optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, 2012.

10 R.M. Lusby, J. Larsen, M. Ehrgott, and D.M. Ryan. A set packing inspired method for real-time junction train routing. *Computers & Operations Research*, 2012.

11 M. Mazzarello and E. Ottaviani. A traffic management system for real-time traffic optimisation in railways. *Transportation Research Part B*, 41:246–274, 2007.

12 J. Pachl. *Railway Operation and Control*. VTD Rail Publishing, Mountlake Terrace, WA, USA, 2002.

13 J. Pachl. Timetable design principles. In I.A. Hansen and J. Pachl, editors, *Railway Timetable & Traffic*, chapter 2, pages 9–42. Eurailpress | DVV Rail Media, Hambourg, Germany, 2008.

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

15 S. Sobieraj, G. Marlière, and J. Rodriguez. Simulation of solutions of a fixed-speed model for the real-time railway traffic optimization problem. In *RailRome 2011, 4th International Seminar on Railway Operations Modelling and Analysis*, Rome, Italy, 2011.

16 J. Törnquist and J.A. Persson. N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B*, 41:342–362, 2007.

17 J. Törnquist Krasemann. Design of an effective algorithm for fast response to re-scheduling of railway traffic during disturbances. *Transportation Research Part C*, 20:62–78, 2012.

18 U.I.C. Leaflet 406 "capacity", 2004.