# A three-step Benders decomposition for the real-time Railway Traffic Management Problem

Kaba Keita [a],[*], Paola Pellegrini [b], Joaquin Rodriguez [a]

[a] Univ Lille Nord de France, IFSTTAR, COSYS, ESTAS F-59000, Lille, F-59650, Villeneuve D'Ascq, France
[b] Univ Lille Nord de France, IFSTTAR, COSYS, LEOST F-59000, Lille, F-59650, Villeneuve D'Ascq, France

ABSTRACT

In railway systems, during congested traffic situations, the infrastructure capacity is completely exploited for trains circulation. When traffic is perturbed, some trains must be stopped or slowed down for ensuring safety, and delays occur. The real-time Railway Traffic Management Problem (rtRTMP) is the problem of modifying trains route and schedule to limit delay propagation. In this paper, we propose an approach based on Benders decomposition of a MILP-based algorithm for this problem, named RECIFE-MILP. Specifically, we split the solution process in three steps rather than two as in the standard decomposition. As we show in a thorough experimental analysis, this decomposition of the solution process into three steps can outperform the original RECIFE-MILP algorithm when tackling large instances with some specific features.

## 1. Introduction

For many railway systems, in congested situations, the infrastructure capacity is completely exploited for train circulations. Many trains travel within short time through critical points. Here, if a disturbance occurs, traffic may be perturbed and, as a result, conflicts may emerge. In a conflict, multiple trains traveling at the planned speed would claim the same track segment concurrently. Hence, some trains must be stopped or decelerated for ensuring safety, and delays propagate. In locations such as junctions, which are areas where multiple lines cross, the emergence of conflicts is very frequent and effectively dealing with them may be particularly difficult. The real-time Railway Traffic Management Problem (rtRTMP) is the problem of modifying trains route and schedule to limit delay propagation. Currently, this problem is tackled by dispatchers. They do it manually, so the result of their choices is generally sub-optimal. The use of an efficient algorithm to help dispatchers decision making is crucial to ensure an effective traffic management when disturbances occur. The design of such an efficient algorithm has been the object of a remarkable branch of research, to which this paper belongs.

In the literature, many algorithms have been proposed to address the rtRTMP. In this paper, we aim to increase the applicability of an existing algorithm. Namely, we consider the RECIFE-MILP algorithm. RECIFE-MILP is a Mixed Integer Linear Programming (MILP) based heuristic which has been proven to be very effective in several circumstances. However, it has been shown (Pellegrini et al., 2015) that the performance of RECIFE-MILP may strongly worsen when tackling large instances in the short time allowed by the real-time nature of the problem. For the MILP formulation at the basis of this algorithm, we propose a Benders Decomposition (BD) (Benders, 1962) approach. In a classic BD, binary variables, here representing train routing and scheduling decisions, are fixed in the

master problem. Given these decisions, the continuous variables, i.e., the trains actual arrival times, are computed in the slave problem to deduce the total delay. Through the progressive addition of cuts to the master and the iterative solution of the two problems, the algorithm is supposed to converge to the optimal solution. Unfortunately, this approach is not able to improve the performance of RECIFE-MILP (Keita, 2017). We think that this bad performance is due to three main issues, deriving from the initial MILP model of RECIFE-MILP. First, the objective function of this model only includes continuous variables, which implies a rather blind master problem at least at the beginning of the search: its objective function includes only a dummy variables representing the contribution of continuous variables, and is hence incapable of driving the search until the addition of a sufficient number of cuts. Second, in RECIFE-MILP the relation between routing and scheduling variables is completely held by the continuous timing variables: they are hence absent in the master problem, which strives in finding feasible solutions for the overall problem for a number of iterations. Third, the RECIFE-MILP model is a disjunctive programming formulation which includes a big-M parameter linked to the scheduling variables, which is hence included in the cuts making them potentially quite weak. While the first issue cannot be solved without fundamentally changing the aim of the RECIFE-MILP model, in this paper we propose an approach for overcoming the second and the third one. In particular, we propose a three-step Benders Algorithm (3BA). Specifically, we split the solution process in three steps rather than two as in the standard BD. In a first step, a master problem including only routing variables is tackled. Based on the fixed routing, in a second step a mixed-integer slave problem finds the optimal scheduling decisions. Finally, in a third step a continuous slave problem computes the delays corresponding to the routing and scheduling, and generates cuts for the master. As we show in a thorough experimental analysis, this decomposition of the solution process into three steps outperforms the original RECIFE-MILP algorithm when tackling large instances with some specific features.

The remainder of this paper is organized as follows. Section 2 presents a literature review. In Section 3, we describe the rtRTMP tackled in the paper and present its mathematical formulation used in RECIFE-MILP. An introduction to the RECIFE-MILP algorithm is done in Section 4. The details on the classic Benders Algorithm (CBA) and the three-step Benders Algorithm (3BA) are described in Sections 5 and 6, respectively. Computational results are reported in Section 7. Conclusions and directions for future work are discussed in Section 8.

## 2. Literature review

Many contributions can be found in the literature which propose algorithms, including decomposition ones, to address the rtRTMP. In the following, we report some relevant contributions. We group the contributions in two categories: those that propose algorithms without focusing on the size of the instances, and those that explicitly tackle large instances through decomposition. For what concerns the first category, several algorithms have been proposed (for recent surveys, see (Cacchiani et al., 2014; Corman and Meng, 2015; Fang et al., 2015)). In some papers, the authors propose solution approaches in which train routes are fixed and the optimization concerns the scheduling decisions ((Sahin, 1999; Dessouky et al., 2006; Törnquist Krasemann, 2015)). Some authors propose solution approaches that consider a limited number of alternative routes selected during the solution process ((D'Ariano et al., 2007; Corman et al., 2010; Acuna-Agost et al., 2011; Samà et al., 2016; Samà et al., 2017); others start with a small number of alternative routes and increase it throughout the solution process (Caimi et al., 2011, 2012). While several algorithms exist to tackle the rtRTMP considering a limited number of alternative routes, few algorithms have been proposed to address the rtRTMP considering all alternative routes that can be traversed given the infrastructure layout. An effort in this direction is the stream of work including (Rodriguez, 2007; Törnquist Krasemann, 2012; Pellegrini et al., 2014, 2015). All these papers do not explicitly discuss the size of the instances to be tackled. Differently, some works exist dealing specifically with large instances and using decomposition techniques. For example, some authors propose heuristic approaches in which the size of the instances is reduced through spacial decomposition of the infrastructure on which traffic is to be managed ((Corman et al., 2012; Corman et al., 2014; Kecman et al., 2013)). Others, as (Lusby et al., 2012; Meng and Zhou, 2014), propose classic mathematical decomposition approaches based on column generation and Lagrangian relaxation, respectively. To the best of our knowledge, only (Lamorgese and Mannino, 2015; Lamorgese et al., 2016; Lamorgese and Mannino, 2016) propose algorithms based on BD to solve the rtRTMP. In a recent seminar (Lamorgese and Mannino, 2016), presents a standard BD approach to tackle the train's rescheduling problem formulated with an alternative graph model. In their approach, the authors replace the standard Benders feasibility and optimality cuts with strong cuts obtained by strengthening and lifting the standard ones.

In this paper, we consider the algorithm named RECIFE-MILP, which allows solving some rtRTMP instances to optimality considering all details in the infrastructure. RECIFE-MILP has been tested on instances representing traffic in various control areas with different characteristics (Pellegrini et al., 2015, 2016; Quaglietta et al., 2016), achieving good results. However, it has been shown that in some circumstances the performance of the algorithm worsens, as in case trains have a large number of re-routing possibilities (Pellegrini et al., 2015).

## 3. The real-time Railway Traffic Management Problem

As mentioned in the introduction, the rtRTMP consists in modifying trains route and schedule to limit delay propagation in case of traffic perturbation.

In principle, planned timetables can be smoothly operated: if there are no disturbances nor disruptions, then the timetable can be executed exactly as planned. However, in real-time operations, disturbances and disruptions are inevitable. Disturbances are relatively small perturbations that influence the railway traffic causing trains' primary delay. They consist for example in a longer than planned running time from one station to another or in a longer dwell time at a stop. These delays often propagate generating the so-called secondary or knock-on delays. Disruptions are large perturbations leading to major modifications of the timetable. The rtRTMP

concerns the management of traffic in case of disturbances.

Several approaches exist for modeling the rtRTMP. The representation considered for infrastructure has a strong impact on the modeling techniques that can be effectively applied. Specifically, macroscopic, mesoscopic and microscopic representations have been considered. In the macroscopic representation, the railway network can be seen as a graph in which stations corresponds to nodes and lines to edges. Precise train dynamics are in general neglected as well as actual train separation. In the mesoscopic representation, some parts of the network are represented in detail, and some others are not. Finally, in the microscopic representation, all details of the infrastructure are modeled. Indeed, when moving from macroscopic to mesoscopic and finally to microscopic representations, the realism of the problem tackled increases. This allows in principle to find solutions that could be directly deployed. This comes at the cost of a larger amount of data necessary to model the same portion of infrastructure, and hence of a typically higher difficulty in finding good solutions to the problem. Here, we focus on the microscopic representation: we aim at efficient algorithms actually able to tackle traffic in interesting portions of the network. We consider the highest level of detail for infrastructure representation, i.e., we consider track-circuits. Track-circuits are track sections on which the presence of a train is automatically detected. Sequences of track-circuits are grouped into block sections, the access to which is controlled by a signal. To provide clear signal to the driver, before a train can enter (start the occupation of) a sequence of block sections, all their track-circuits must be reserved for the train itself, also allowing some additional time for route formation. After a train exits a track-circuit, its reservation is still active for the so called release time. This functioning precisely mimics reality, and in particular the *route-lock sectional-release* interlocking system (Pachl et al., 2009). We name utilization the sum of reservation and occupation time. The complete sequence of track-circuits traversed by a train during its trip is named route. Routes may include intermediate stops. In the representation we consider, a sequence of track-circuits used by trains either performing stops or not defines two routes. The running times for the route with intermediate stops includes the appropriate deceleration and acceleration times.

The objective functions used in literature when tackling the rtRTMP are typically functions of trains' delay, e.g., total delay or maximum delay. Trains' delay is the non-negative difference between the actual arrival times at stations and the scheduled ones. Other objective functions considering different performance indicators are: the minimization of the trains' travel time, which is a measure of the time spent by all trains in the network, the schedule deviation, which penalizes both trains' advance and delay, and the maximization of trains' punctuality, which refers to the number of trains that do not arrive at their final destination on time, or suffering a delay smaller than a given threshold. Moreover, customer perspective is sometimes considered when solving the rtRTMP. For example, passengers' inconvenience to be minimized can be measured as the additional waiting time at platforms and number of transfers.

The rtRTMP studied in this paper minimizes weighted total delay. The weighting in the objective function allows taking into account different train priorities. These priorities may be linked to the type of circulation (e.g., freight or passenger) or other aspects as the number of passengers traveling on each train.

Concerning the rtRTMP constraints, the following ones are imposed to respect operational requirements:

**Time concerning constraints:** These constraints impose the respect of operational requirements related to the time. For instance, at a station a train cannot depart before its scheduled departure time. Another requirement handled by these constraints is the coherent physical occupation time of track-circuits along a train route: a train cannot start occupying a track-circuit along its route if it has not spent in the preceding one at least the corresponding running time. Time constraints also allow modeling the complete blocking time stairway (Pachl et al., 2009).

**Connection constraints:** In railway traffic, train services are sometimes in connection: passengers or crew members are planned to transfer from a train to another at a specific station. In this case we must impose constraints that manage requirements such as the guarantee of a minimum time separation between the arrival of the feeder train and the departure of the connecting train.

**Rolling-stock re-utilization constraints:** If train services are to be operated with the same rolling-stock, their arrival and departure must be coherent, in terms of both time and space. Hence, a minimum time interval must pass between arrival and departure, and the concerned trains must arrive at and depart from the same track-circuit.

**Capacity constraints:** The track-circuits utilization by two trains must not overlap.

Modeling this problem, we present in the following the RECIFE-MILP formulation, which we decompose in our Benders algorithms. Before presenting the formulation, we present the notation used, following the notation in (Pellegrini et al., 2015).

$T \equiv$ set of trains;

$w_t \equiv$ weight associated to train $t$'s delay;

$ty_t \equiv$ type corresponding to train $t$ (train characteristics);

$init_t, exit_t \equiv$ earliest time at which train $t$ can be operated and earliest time at which it can reach its destination given $init_t$ and the route assigned in the timetable;

$R_t, TC_t \equiv$ set of routes and track-circuits available for train $t$;

$TC^r \equiv$ set of track-circuits composing route $r$;

$TC(tc, tc', r) \equiv$ set of track-circuits between $tc$ and $tc'$ along $r$;

$p_{r,tc}, s_{r,tc} \equiv$ track-circuits preceding and following $tc$ along $r$;

$tc_0, tc_\infty \equiv$ dummy track-circuits representing origin and destination of any route;

$rt_{ty,r,tc}, ct_{ty,r,tc} \equiv$ running time and clearing time of $tc$ along $r$ for a train of type $ty$;

$bs_{r,tc} \equiv$ block section including track-circuit $tc$ along route $r$;

$for_{bs}, rel_{bs} \equiv$ formation time and release time for block section $bs$;

$\widehat{TC}_{t,t',tc} \equiv$ set of track-circuits $tc'$ which may be used by both t and $t'$ such that if t precedes ($\prec$) $t'$ on $tc$, then necessarily $t \prec t'$ on $tc'$, and so on (e.g., if the track-circuits follow each other on a straight track segment). In other words, it is the set of track-circuits

composing the track segment which starts with $tc$. $\widehat{TC}_{t,t',tc} = \{tc\}$ if $tc \in TC_t \cap TC_{t'}$ and no implied precedence relation links $tc$ to other track-circuits. $\widehat{TC}_{t,t',tc} = \varnothing$ if $\exists tc' \in TC_t \cap TC_{t'}$ such that $tc \in \widehat{TC}_{t,t',tc'}$, i.e., that is $tc$ belongs to a track segment which starts with a different $tc'$. Hence, each track-circuit belongs to one and only one set $\widehat{TC}_{t,t',tc}$;

$\widehat{TC}_{t,t'} \equiv \cup_{tc \in TC_t \cap TC_{t'}} \widehat{TC}_{t,t',tc}$ , i.e., $\widehat{TC}_{t,t'}$ is the set of all the track segments for the two trains;

$M \equiv$ large constant.

The formulation contains non-negative continuous variables:

for all triplets of $t \in T$, $r \in R_t$ and $tc \in TC^r$:

$o_{t,r,tc}$ : time at which t starts the occupation time of $tc$ along r;

$l_{t,r,tc}$ : longer stay of t's head on $tc$ along r, due to dwell time and scheduling decisions (delay);

for all pairs of $t \in T$ and $tc \in TC_t$:

$sU_{t,tc}$ : time at which t starts $tc$ utilization;

$eU_{t,tc}$ : time at which t ends $tc$ utilization;

for all $t \in T$:

$D_t$ : delay suffered by train t when exiting the infrastructure.

The continuous variables are represented in Fig. 1, together with the main input data (clearing time, running time, release time and formation time) concerning block sections and track-circuits.

In addition it includes **binary variables**:

for all pairs of $t \in T$ and $r \in R_t$:

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

for all triplets of $t, t' \in T$ such that the index of t is smaller than the index of t', and $\widehat{tc} \in \widehat{TC}_{t,t'}$:

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

We impose the index of t smaller than the index of t' to avoid creating two re-scheduling variables $y_{t,t',\widehat{tc}}$ and $y_{t',t,\widehat{tc}}$ for track segment $\widehat{tc}$. One variable is enough in this case since $y_{t',t,\widehat{tc}}$ is the complement to one of $y_{t,t',\widehat{tc}}$. By doing so, we reduce the number of $y$-variables in the formulation.

$$\min \sum_{t \in T} w_t D_t. \tag{1}$$

$$o_{t,r,tc} \geq init_t \, x_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r : p_{r,tc} = tc_0, \tag{2}$$

$$o_{t,r,tc} \leq M x_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r, \tag{3}$$

$$o_{t,r,tc} = o_{t,r,p_{r,tc}} + l_{t,r,p_{r,tc}} + rt_{r,ty,p_{r,tc}} x_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r : tc \neq tc_0, \tag{4}$$

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T, \tag{5}$$

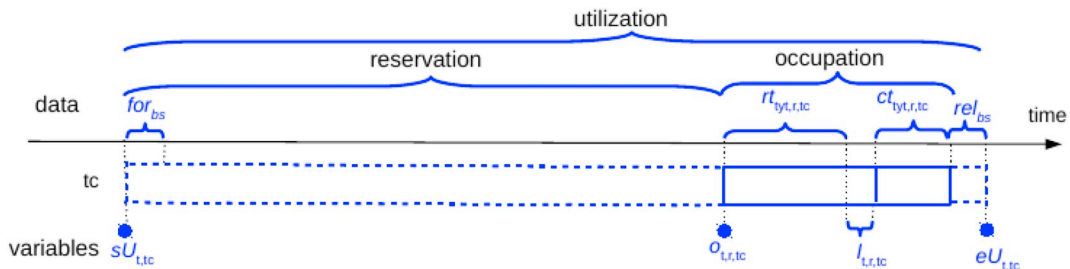$$D_t \geq \sum_{r \in R_t} o_{t,r,tc_\infty} - exit_t \quad \forall t \in T, \tag{6}$$



**Fig. 1.** Main variables and data concerning the utilization of $tc$ belonging to route $r$ of train $t$.

$$sU_{t,tc} \leq \sum_{\substack{r \in R_t: \\ tc \in TC^r}} \left( o_{t,r,ref_{r,tc}} - for_{bs_{r,tc}} \, x_{t,r} \right) \quad \forall t \in T, \, tc \in TC_t, \tag{7}$$

$$eU_{t,tc} \geq \sum o_{t,r,ref_{r,tc}} \begin{array}{c} r \in R_t : \\ tc \in TC^r \end{array} + rel_{bs_{r,tc}} x_{t,r} + ct_{r,ty,tc} x_{t,r} +$$
$$+ \sum_{tc' \in TC\left(ref_{r,tc},tc,r\right)} rt_{r,ty,tc'} x_{t,r} + l_{t,r,tc'} \quad \forall t \in T, tc \in TC_t, \tag{8}$$

$$eU_{t,tc} - M\left(1 - y_{t,t',\widehat{tc}}\right) \leq sU_{t',tc} \quad \forall t, t' \in T, \, t < t', tc, \widehat{tc} \in TC_t \cap TC_{t'} : tc \in \widehat{TC}_{t,t',\widehat{tc}}, \tag{9}$$

$$eU_{t',tc} - My_{t,t',\widehat{tc}} \leq sU_{t,tc} \quad \forall t, t' \in T, \, t < t', tc, \widehat{tc} \in TC_t \cap TC_{t'} : tc \in \widehat{TC}_{t,t',\widehat{tc}}. \tag{10}$$

The objective (1) is the minimization of the total weighted delays suffered by trains at their exit from the infrastructure. Constraints (2) state that a train $t$ cannot be operated earlier than $init_t$. Constraints (3) indicate that the start time of track-circuit occupation along a route is zero if the route itself is not used. Constraints (4) impose that a train starts occupying track-circuit $tc$ along a route after spending in the preceding track-circuit its longer stay and its running time, if the route is used. Constraints (5) state that a train must use exactly one route. Constraints (6) indicate that the value of delay $D_t$ at least equals the difference between the actual and the scheduled arrival times at the exit of the infrastructure. Constraints (7) impose that a train's utilization of a track-circuit $tc$ starts as soon as the train starts occupying the track-circuit $ref_{r,tc}$ along one of the routes including $tc$, minus the formation time. Constraints (8) indicate that the utilization of a track-circuit lasts till the train utilizes it along any route, plus the formation time and the release time. Constraints (9) and (10) impose that the track-circuit utilization by two trains does not overlap.

## 4. RECIFE-MILP algorithm

In this paper, we propose a variant of the RECIFE-MILP algorithm. RECIFE-MILP is a heuristic algorithm which mainly consists in tackling a rtRTMP instance by solving the formulation presented in Section 3, for a limited computational time through a MILP solver. If the optimum is proven, the run is interrupted, otherwise, the best solution found after the elapse of this time is returned. In RECIFE-MILP, a two-optimization-step cycle is implemented. In the first step, the MILP solver optimizes the train scheduling without modifying the routes with respect to the default ones (i.e., the routes defined in the initial timetable). In the second step, it optimizes also in terms of train routing, using as initial solution the best solution found in the first step. Disregarding the optimality proof, the first step is terminated after 30 s provided that one feasible solution has been found. Otherwise, the first step continues until the first feasible solution is detected; in all experiments run so far, the search with no re-routing (first optimization step) always found at least a feasible solution within very few seconds.

Besides the two-optimization-step cycle, others boosting methods are implemented to improve the solution algorithm performance. Among them there are the infrastructure topology exploitation and the decrease of the value of the large constant (big-M decrease).

The infrastructure topology exploitation consists in reducing as much as possible the number of binary $y$-variables, i.e., the variables appearing in Constraints (9) and (10) that define the precedence relation between couples of trains utilizing the same track-circuit. To this aim, the RECIFE-MILP algorithm exploits the fact that the topology of a physical network frequently imposes that the precedence relation between couples of trains must be identical on different track-circuits.

To decrease the value of the large constant (i.e., the big-M value), the solution obtained in the first optimization step is exploited. We refer the interested reader to (Pellegrini et al., 2015) for more details on boosting methods.

## 5. Classic Benders Algorithm (CBA)

In this section, we describe CBA. Here, the original formulation is decomposed into two problems, the Unrestricted Master Problem (UMP) and the Sub-Problem (SP). The UMP includes all binary variables, corresponding to re-routing and rescheduling decisions. The SP includes all variables related to timings. Fig. 2 shows the procedure behind the iterative algorithm. In a first step, the UMP fixes
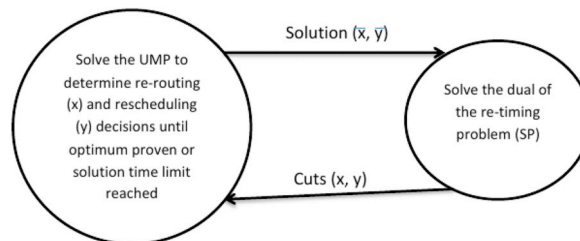


**Fig. 2.** Representation of the classic Benders approach applied to the RECIFE-MILP formulation.

routing and scheduling decisions and passes them to the SP. In a second step, the dual of the latter finds out if timings can be fixed so that a feasible solution is found with the routing and scheduling decisions received. If so, it sends to the UMP an optimality cut to indicate the quality of the decisions received. Otherwise, it returns feasibility cuts aiming to drive the UMP towards feasible routing and scheduling combinations. These two steps are repeated until the optimal solution is found or until a computational time limit is reached. The details on the UMP and the SP are described in the rest of this section.

## 5.1. Unrestricted Master Problem (UMP)

In CBA, the UMP includes both the re-routing variables $x_{t,r}$ and the rescheduling ones $y_{t,t',\widehat{tc}}$, and a dummy non-negative continuous variables $z$. Its formulation is a follows.

$$\min z. \tag{11}$$

Subject to the routing constraints

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T. \tag{12}$$

and the integrality constraints on re-routing and rescheduling variables

$$x_{t,r} \in \{0,1\} \quad \forall t \in T, r \in R_t \tag{13}$$

$$y_{t,t',\widehat{tc}} \in \{0,1\} \quad \forall t, t' \in T, t \prec t', \widehat{tc} \in \widehat{TC}_{t,t'} \tag{14}$$

Constraints (12) state that a train must use exactly one route.

## 5.2. Sub-problem (SP)

Let, $\bar{r} \in R_t$ be the route used by train $t$ and $\bar{\bar{r}} \in R_{t'}$ the route used by train $t'$ in the UMP solution to be evaluated, where $t, t' \in T$. Remark that the set of track-circuits available for train $t$ ($TC_t$) can be reduced to $TC^{\bar{r}}$ (set of track-circuits composing route $\bar{r}$) and the set of track-circuits available for train $t'$ ($TC_{t'}$) can also be reduced to $TC^{\bar{\bar{r}}}$. By excluding all other track-circuits, for which timing variables can be implicitly set to zero, we minimize the size of the SP. Hence, the SP can then be formulated as follows.

$$\min \sum_{t \in T} w_t D_t. \tag{15}$$

$$o_{t,\bar{r},tc} \geq init_t \quad \forall t \in T, tc \in TC^{\bar{r}} : p_{\bar{r},tc} = tc_0 \tag{16}$$

$$-o_{t,\bar{r},tc} + o_{t,\bar{r},p_{r,tc}} + l_{t,\bar{r},p_{\bar{r},tc}} = -rt_{\bar{r},ty,p_{\bar{r},tc}} \quad \forall t \in T, tc \in TC^{\bar{r}} : tc \neq tc_0 \tag{17}$$

$$D_t - o_{t,\bar{r},tc_\infty} \geq -exit_t \quad \forall t \in T \tag{18}$$

$$sU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} \leq -for_{bs_{\bar{r},tc}} \quad \forall t \in T, tc \in TC^{\bar{r}} \tag{19}$$

$$eU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} - \sum_{tc' \in TC\left(ref_{\bar{r},tc}, tc, \bar{r}\right)} l_{t,\bar{r},tc'} \geq rel_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} +$$
$$+ \sum_{tc' \in TC\left(ref_{\bar{r},tc}, tc, \bar{r}\right)} rt_{\bar{r},ty,tc'} \quad \forall t \in T, tc \in TC^{\bar{r}} \tag{20}$$

$$eU_{t,tc} - sU_{t',tc} \leq M - M\bar{y}_{t,t',\widehat{tc}} \quad \forall t, t' \in T, t < t', tc, \widehat{tc} \in TC^{\bar{r}} \cap TC^{\bar{\bar{r}}} : tc \in \widehat{TC}_{t,t',\widehat{tc}} \tag{21}$$

$$eU_{t',tc} - sU_{t,tc} \leq M\bar{y}_{t,t',\widehat{tc}} \quad \forall t, t' \in T, t < t', tc, \widehat{tc} \in TC^{\bar{r}} \cap TC^{\bar{\bar{r}}} : tc \in \widehat{TC}_{t,t',\widehat{tc}}. \tag{22}$$

The optimality and feasibility cuts we add to the UMP during Benders iterations and obtained from the dual of the SP are the following:

$$
\begin{aligned}
z - \Bigg( & \sum_{t\in T}\sum_{tc\in TC^{\bar{r}}} x_{t,\bar{r}}\Big( init_t\alpha_{t,\bar{r},tc} - rt_{\bar{r},ty,p_{\bar{r},tc}}\lambda_{t,\bar{r},tc}\Big) + \\
& +\sum_{t\in T}\sum_{tc\in TC^{\bar{r}}} x_{t,\bar{r}}\Bigg[\Bigg( rel_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \sum_{tc'\in TC\left(ref_{\bar{r},tc},tc,\bar{r}\right)} rt_{\bar{r},ty,tc'}\Bigg)\omega_{t,tc} - for_{bs_{\bar{r},tc}}\varphi_{t,tc}\Bigg] - \\
& -\sum_{t\in T} exit_t\theta_t + \sum_{t,t'\in T}\sum M^{\widehat{tc},\, tc\in TC^{\bar{r}}\cap TC^{\bar{r}'}}_{\quad tc\in \widehat{TC}_{t,t',\widehat{tc}}} : \Bigg[\Big(1-y_{t,t',\widehat{tc}}\Big)\eta_{t,t',tc} + y_{t,t',\widehat{tc}}\psi_{t,t',tc}\Bigg]\Bigg) \geq 0 \\
& ((\alpha,\lambda,\theta,\varphi,\omega,\eta,\psi)\in P_S)
\end{aligned}
\tag{23}
$$

$$
\begin{aligned}
& \sum_{t\in T}\sum_{tc\in TC^{\bar{r}}} x_{t,\bar{r}}\Big( init_t\alpha_{t,\bar{r},tc} - rt_{\bar{r},ty,p_{\bar{r},tc}}\lambda_{t,\bar{r},tc}\Big) \\
& +\sum_{t\in T}\sum_{tc\in TC^{\bar{r}}} x_{t,\bar{r}}\Bigg[\Bigg( rel_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \sum_{tc'\in TC\left(ref_{\bar{r},tc},tc,\bar{r}\right)} rt_{\bar{r},ty_t,tc'}\Bigg)\omega_{t,tc} - for_{bs_{\bar{r},tc}}\varphi_{t,tc}\Bigg] - \\
& -\sum_{t\in T} exit_t\theta_t + \sum_{t,t'\in T}\sum M^{\widehat{tc},\, tc\in TC^{\bar{r}}\cap TC^{\bar{r}'}}_{\quad tc\in \widehat{TC}_{t,t',\widehat{tc}}} : \Big[(1-y_{t,t',\widehat{tc}})\eta_{t,t',tc} + y_{t,t',\widehat{tc}}\psi_{t,t',tc}\Big] \leq 0 \\
& ((\alpha,\lambda,\theta,\varphi,\omega,\eta,\psi)\in R_S)
\end{aligned}
\tag{24}
$$

where $\alpha_{t,\bar{r},tc}, \lambda_{t,\bar{r},tc}, \theta_t, \varphi_{t,tc}, \omega_{t,tc}, \eta_{t,t',tc}, \psi_{t,t',tc}$, be the dual variables associated with Constraints (16)–(22) respectively. As such, $\alpha_{t,\bar{r},tc} \geq 0, \lambda_{t,\bar{r},tc}\in\mathbb{R}, \theta_t\geq 0, \varphi_{t,tc}\leq 0, \omega_{t,tc}\geq 0, \eta_{t,t',tc}\leq 0$ and $\psi_{t,t',tc}\leq 0$.

## 6. Three-step Benders Algorithm (3BA)

Inspired by CBA, we design 3BA and we describe it in this section. 3BA separates the RECIFE-MILP formulation into three problems: the Unrestricted Master Problem (UMP), the Mixed Integer Linear Sub-Problem (MILSP) and the Linear Sub-Problem (LSP). In 3BA, the UMP contains only the binary re-routing variables. The MILSP contains the binary rescheduling variables and the continuous timing variables. The Linear Sub-Problem (LSP) contains only timing variables, as the SP in CBA. Fig. 3 shows the procedures behind the algorithm. As for CBA, 3BA is iterative. In a first step, the UMP fixes routing decisions and passes them to the MILSP. In a second step, the MILSP solves the scheduling problem finding the optimal train precedences given the routes received. These routes and precedences are passed to the LSP. In a third step, the dual of the LSP determines the optimality cuts to be sent to the UMP to represent the quality of the routing decisions. Remark that, differently from CBA, here cuts include no information on precedence relations among trains. These three steps are repeated in sequence until the optimal solution is found or until a computational time limit is reached. The details on the UMP, the MILSP and the LSP are described in the rest of this section.

### 6.1. Unrestricted Master Problem (UMP)

Unlike in CBA, the UMP in 3BA includes only the re-routing variables $x_{t,r}$ and a dummy non-negative continuous variables $z$. Its formulation is a follows.

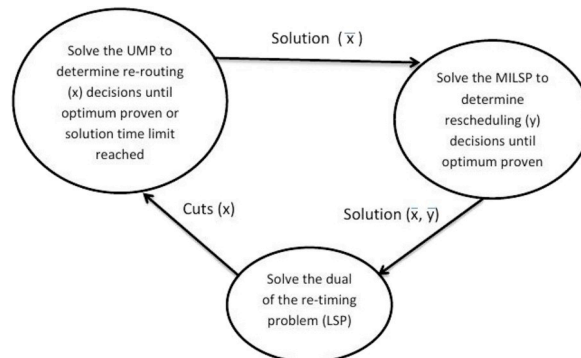$$\min z. \tag{25}$$

Subject to the routing constraints



**Fig. 3.** Representation of the three-step Benders Algorithm applied to the RECIFE-MILP formulation.

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T \tag{26}$$

and the integrity constraints on re-routing variables

$$x_{t,r} \in \{0,1\} \quad \forall t \in T, r \in R_t \tag{27}$$

Constraints (26) state that a train must use exactly one route.

## 6.2. Mixed integer linear Sub-Problem (MILSP)

When the re-routing variables are fixed in the UMP (let them be $\bar{x}_{t,r}$), then we know the route used, $\bar{r} \in R_t$, by each train $t$. As in CBA, the set of track-circuits available for train $t$ ($TC_t$) can be reduced to $TC^{\bar{r}}$ (set of track-circuits composing route $\bar{r}$) and the set of track-circuits available for train $t'$ ($TC_{t'}$) can also be reduced to $TC^{\bar{r}}$. The MILSP can then be formulated as follows.

$$\min \sum_{t \in T} w_t D_t. \tag{28}$$

Subject to the following constraints:

$$o_{t,\bar{r},tc} \geq init_t \quad \forall t \in T, tc \in TC^{\bar{r}} : p_{\bar{r},tc} = tc_0 \tag{29}$$

$$-o_{t,\bar{r},tc} + o_{t,\bar{r},p_{\bar{r},tc}} + l_{t,\bar{r},p_{\bar{r},tc}} = -rt_{\bar{r},ty,p_{\bar{r},tc}} \quad \forall t \in T, tc \in TC^{\bar{r}} : tc \neq tc_0 \tag{30}$$

$$D_t - o_{t,\bar{r},tc_\infty} \geq -exit_t \quad \forall t \in T \tag{31}$$

$$sU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} \leq -for_{bs_{\bar{r},tc}} \quad \forall t \in T, \ tc \in TC^{\bar{r}} \tag{32}$$

$$eU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} - \sum_{tc' \in TC\left(ref_{\bar{r},tc},tc,\bar{r}\right)} l_{t,\bar{r},tc'} \geq rel_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} +$$
$$+ \sum_{tc' \in TC\left(ref_{\bar{r},tc},tc,\bar{r}\right)} rt_{\bar{r},ty,tc'} \quad \forall t \in T, tc \in TC^{\bar{r}} \tag{33}$$

$$eU_{t,tc} - sU_{t',tc} \leq M - My_{t,t',\widehat{tc}} \quad \forall t,t' \in T, \ t < t', tc, \widehat{tc} \in TC^{\bar{r}} \cap TC^{\bar{r}} : tc \in \widehat{TC}_{t,t',\widehat{tc}} \tag{34}$$

$$eU_{t',tc} - sU_{t,tc} \leq My_{t,t',\widehat{tc}} \quad \forall t,t' \in T, \ t < t', tc, \widehat{tc} \in TC^{\bar{r}} \cap TC^{\bar{r}} : tc \in \widehat{TC}_{t,t',\widehat{tc}}. \tag{35}$$

Note that in the MILSP, Constraints (34) and (35) contain the binary variables $y_{t,t',\widehat{tc}}$. Hence, we cannot formulate the dual of this problem and identify the cut to be added to the UMP. To do so, we solve the MILSP to get the optimal rescheduling decisions $\bar{y}_{t,t',\widehat{tc}}$, then we construct the LSP given $\bar{y}_{t,t',\widehat{tc}}$ and the re-routing decisions $\bar{x}_{t,r}$ fixed in the UMP. The formulation of the LSP is as follows.

## 6.3. Linear Sub-Problem (LSP)

Indeed, when we know the re-routing decisions made in the UMP (let them be $\bar{x}_{t,r}$) and the corresponding rescheduling decisions made in the MILSP (let them be $\bar{y}_{t,t',\widehat{tc}}$), the remaining problem is a re-timing one. The LSP is formulated as the MILSP with the exception of Constraints (34) and (35). Here, according to the value of the rescheduling decisions $\bar{y}_{t,t',\widehat{tc}}$ made in the MILSP we add to the LSP one of the following two sets of constraints:

$$eU_{t,tc} - sU_{t',tc} \leq 0 \quad \forall t,t' \in T, \ t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}} \tag{36}$$

$$eU_{t',tc} - sU_{t,tc} \leq 0 \quad \forall t,t' \in T, \ t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}}. \tag{37}$$

If $\bar{y}_{t,t',\widehat{tc}} = 1$ then we add Constraints (36) to the LSP, otherwise (37) is added.

Indeed, the MILSP and the LSP find the same values for timing decisions. However, solving the dual of the LSP allows the addition of cuts to the UMP containing only re-routing variables $x$.

Let $\alpha_{t,\bar{r},tc}, \lambda_{t,\bar{r},tc}, \theta_t, \varphi_{t,tc}, \omega_{t,tc}, \eta_{t,t',tc}, \psi_{t,t',tc}$, be the dual variables associated with Constraints (29)–(33) and (36)–(37) respectively. As such, $\alpha_{t,\bar{r},tc} \geq 0, \lambda_{t,\bar{r},tc} \in \mathbb{R}, \theta_t \geq 0, \varphi_{t,tc} \leq 0, \omega_{t,tc} \geq 0, \eta_{t,t',tc} \leq 0$ and $\psi_{t,t',tc} \leq 0$. We formulate the DLSP that is necessary to formulate the Benders cuts to be added to the UMP.

As the MILSP finds the optimal schedule given the routing decisions, the re-timing problem with such schedule and routing is feasible by definition. Thus, the DLSP is always bounded, and all Benders cuts returned are optimality ones. Remark that, as neither the UMP nor the DLSP include $y$-variables, these cuts are formulated as follows:

$$
z - \left( \sum_{t \in T} \sum_{\bar{r} \in R_t} \sum_{tc \in TC^{\bar{r}}} x_{t,\bar{r}} \left( init_t \alpha_{t,\bar{r},tc} - rt_{\bar{r},ty,p_{\bar{r},tc}} \lambda_{t,\bar{r},tc} \right) + \right.
$$

$$
+ \sum_{t \in T} \sum_{\bar{r} \in R_t} \sum_{tc \in TC^{\bar{r}}} x_{t,\bar{r}} \left[ \left( rel_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \sum_{tc' \in TC\left(ref_{\bar{r},tc,\bar{r}}\right)} rt_{\bar{r},ty,tc'} \right) \omega_{t,tc} - for_{bs_{\bar{r},tc}} \varphi_{t,tc} \right] -
$$

$$
\left. - \sum_{t \in T} exit_t \theta_t \right) \geq 0 \tag{38}
$$

$$
((\alpha, \lambda, \theta, \varphi, \omega) \in P_S)
$$

with $P_S$ the extreme points of the polyhedron $S$ representing the feasible solution space of the DLSP.

## 7. Computational analysis

In this section, we compare the performance of 3BA, CBA and RECIFE-MILP. The implementation is done using IBM ILOG CPLEX Concert Technology for C++ (IBM ILOG CPLEX version 12.7 (IBM, 2017)).

We consider four case studies representing traffic in four control areas in France: the Pierrefitte-Gonesse junction (Gonesse), the Saint-Lazare station (St. Lazare), the Lille Flandres station (Lille) and a line section around the Rouen-Rive-Droite station (Rouen). We set the computational time available for the optimization to 3 min as often done in the literature (e.g. (Pellegrini et al., 2015; Samà et al. 2015).

The weight associated to delay in the objective function is set to 1 for all trains, since we have no information on different train priorities. The route formation and release time are 15 and 5 s for all block section, respectively.

For each control area, we consider a one-day timetable, and we create 100 scenarios in which 20% of trains, randomly selected, suffer a random delay between 5 and 15 min at their entrance in the control area. We generate one rtRTMP instance from each of these 100 scenarios by considering all the trains entering the control area within a 1-h horizon. We set the time horizon from 6:00 a.m. to 07:00 a.m. This time horizon corresponds to the morning peak hour. We use the so obtained 400 instances for testing the performance of 3BA.

Before presenting the results, we describe each control area in the next subsection.

### 7.1. Control areas tackled

**Pierrefitte–Gonesse junction (Gonesse)**: Gonesse is a critical control area north of Paris with dense mixed traffic. Fig. 4 schematically depicts its infrastructure. It includes 89 track-circuits, grouped into 174 block sections and 39 routes. The 1-h instances we tackle include between 14 and 17 trains (mean 15). Each train can use between 5 and 13 routes (mean 8), which translates into a RECIFE-MILP formulation with about 121 000 continuous variables, 1 800 binary variables and 35 000 constraints for an instance with 15 trains.

**Lille Flandres station (Lille)**: In Lille, trains must necessarily cross for entering and exiting the station, which makes even the only scheduling decisions extremely critical. In addition to this difficulty for schedule optimization, the large number of routes available for the trains increases the complexity of the routing optimization w.r.t. Gonesse. The control area is shown in Fig. 5. Lille is a terminal station with 17 platforms connected to seven regional, national and international lines. It hosts both traditional passenger trains and high speed ones. In total, 299 track-circuits compose 734 block sections and 2409 routes. The 1-h instances that we tackle include between 36 and 41 trains (mean 39). Each train can use between 1 and 72 routes (mean 11), which translates into a RECIFE-MILP formulation with about 274 500 continuous variables, 3 700 binary variables and 68 700 constraints, for an instance with 39 trains.

**Saint-Lazare station (St. Lazare)**: St. Lazare is one of Paris complex and critical control areas. As Lille, it is a terminal station area. It covers slightly more than 4 km. The control area is depicted in Fig. 6. The Saint-Lazare station includes 212 track-circuits, grouped into 197 block sections and 27 routes. The 1-h instances tackled include between 54 and 64 trains (mean 60). Each train can use
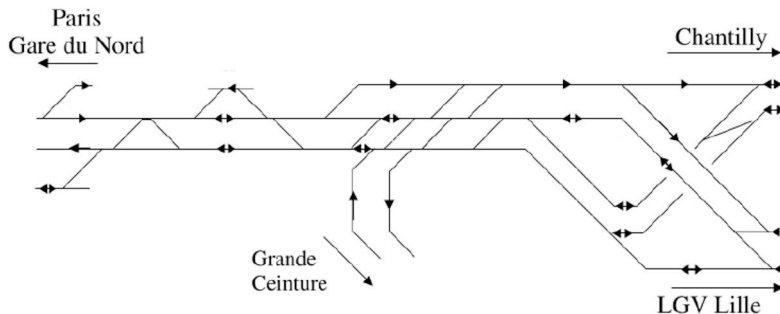
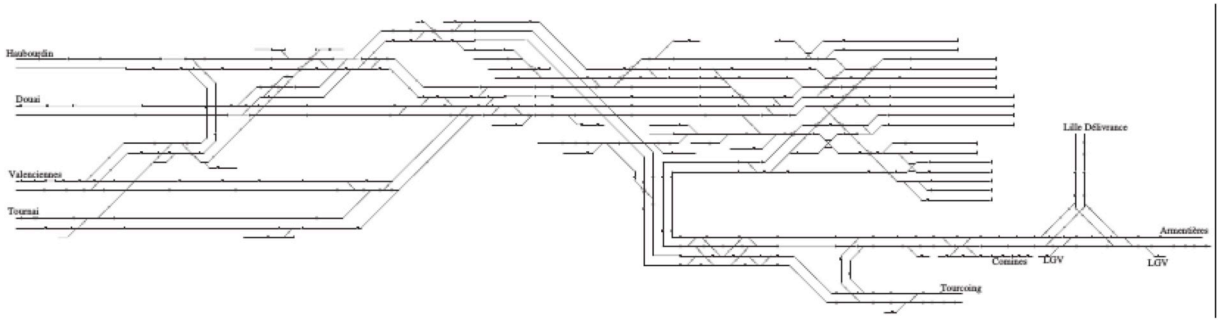

**Fig. 4.** Pierrefitte–Gonesse junction.
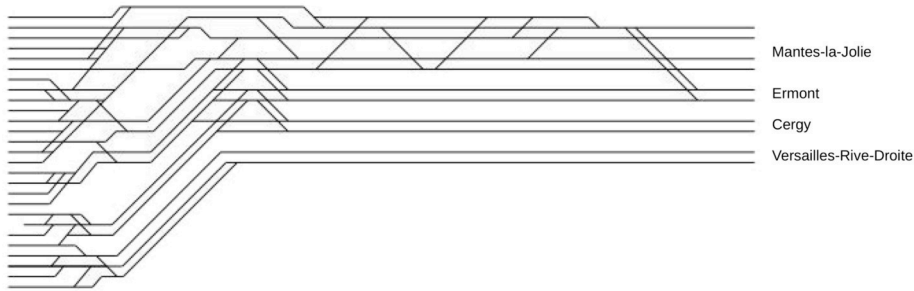
**Fig. 5.** Lille Flandres station.



**Fig. 6.** Saint–Lazare station.

between 1 and 9 routes (mean 5), which translates into a RECIFE-MILP formulation with about 165 016 continuous variables, 4 436 binary variables and 48 655 constraints for an instance with 60 trains.

**Rouen-Rive-Droite control area (Rouen):** The control area including Rouen-Rive-Droite station comprises six stations, with two to six platforms, and one junction. The presence of multiple stations with several possible platform assignments implies the availability of a very large number of alternative routes. The existence of these routes is the main source of complexity of traffic management in this control area, together with the presence of mix traffic. The control area is depicted in Fig. 7. The 190 track-circuits compose 189 block sections and 11 347 routes. Our 1-h instances include between 10 and 13 trains (mean 11). Each train can use between 1 and 384 routes (mean 73), which translates into a RECIFE-MILP formulation with about 949 800 continuous variables, 1030 binary variables and 224 900 constraints, for an instance including 11 trains.

### 7.2. Results

In this section, we present the results of our computational experiments on the four control areas. The aim of these computational experiments is twofold: first, we show and analyze the out-performance of 3BA with respect to CBA; then, we evaluate and compare the performance of 3BA and RECIFE-MILP.

Table 1 reports the results obtained over the 100 instances for each control area after 180 wall-clock seconds of computation. The table contains three columns, one per algorithm. Each column indicates the average delay in seconds (objective function value), the number of optima proven, the average optimality gap found by the corresponding algorithm and the average computational time in seconds. Recall that this time is at most 180 s, i.e., the time limit we consider, but it can be lower if the optimal solution is proven earlier. Four rows make up the table, one for each control area tackled.

The comparison of the performance of 3BA and CBA allows making the following observation. The average returned objective function value of 3BA is lower than the one of CBA for three control areas (Gonesse, Lille and Rouen), while the number of optimal solutions proven is higher for two control areas (Gonesse and Rouen). For the St. Lazare instances, the two algorithms achieve similar performances. In the computational time available, the number of cuts generated for 3BA is lower than the one for CBA. In particular,
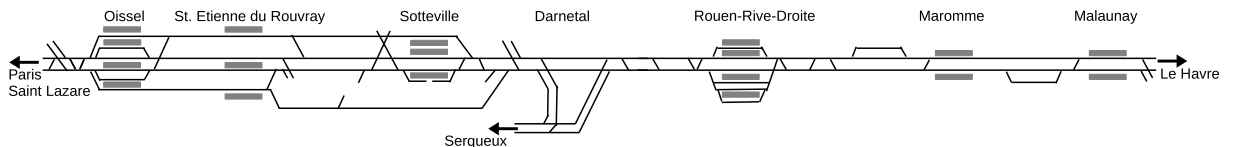


**Fig. 7.** Rouen-Rive-Droite control area.

**Table 1**
Results over 100 instances at each control area: average total delay in seconds, number of optima proven, average optimality gap and average solution time in seconds.

| Control area | Algorithm | | | | | | | | | | | |
| | CBA | | | | 3BA | | | | RECIFE-MILP | | | |
| | Delay | #Opt | Gap | time | Delay | # Opt | Gap | time | Delay | # Opt | Gap | time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Gonesse | 134 | 53 | 0.14 | 87 | 131 | 56 | 0.11 | 80 | 104 | 100 | 0 | 64 |
| Lille | 1146 | 0 | 1 | 180 | 1016 | 0 | 0.97 | 180 | 567 | 15 | 0.53 | 154 |
| St. Lazare | 447 | 0 | 1 | 180 | 447 | 0 | 1 | 180 | 176 | 85 | 0.03 | 108 |
| Rouen | 445 | 2 | 0.98 | 180 | 24 | 76 | 0.24 | 44 | 387 | 6 | 0.92 | 169 |

CBA generates 96, 211, 116 and 287 cuts in average versus 93, 129, 75, and 94 for 3BA on the Gonesse, St. Lazare, Lille and Rouen instances, respectively. The lower number for 3BA can be explained by the longer time needed for the generation of one cut, which includes the solution of a MILP problem. We think the better performance of 3BA with respect to CBA is due to three reasons. First, in 3BA no big-M is present in the cuts. This makes the cuts stronger, since big-M is known to be a source of weakness in integer programming. Second, in 3BA, the cuts associated to routing solutions in the UMP are always linked to optimal scheduling decisions. This can be interpreted as having the cuts really indicating what best can be done with the routes selected. In CBA, instead, with a set of routes comes a set of scheduling decisions, which can be good or bad if at all feasible. The cut added to the UMP is then not necessarily linked to the actual routes potential. Third, in CBA several feasibility cuts are generated before the UMP is able to find a feasible solution. Feasibility cuts are indeed quite weak, and this process consumes computational time. On the contrary, in 3BA, only optimality cuts are added, since solutions are by definition feasible.

These results show that 3BA outperforms CBA whatever the chosen criterion (average delay, number of optimal solutions, average optimality gap, average computational time). In the following, then, we only focus on 3BA in the comparison with RECIFE-MILP.

The comparison between 3BA and RECIFE-MILP is shown in Table 1. The results show that 3BA clearly outperforms RECIFE-MILP in Rouen instances. 3BA solves 76 instances to the optimum, finds 24 s as the average delay and 0.24 as the average optimality gap. The corresponding figures for RECIFE-MILP are 6, 387 and 0.92, respectively. However, in the Gonesse, Lille and St. Lazare instances, the performance of 3BA is not comparable to the one of RECIFE-MILP.

To find an explanation to the different relative performance in the control areas, we conjecture that 3BA is appropriate for the instances in which the number of alternative routes available per train is very large. To support our conjecture, we study in the following how the results on each control area change when we vary the maximum number of alternative routes per train. To do so, we solve the same instances considered above limiting the number of routes per train to different values. Specifically, we consider the first $m$ available routes for each train, and we vary $m$ between 1 and the maximum number of routes in the original instances.

The graphical analysis of these results is done in Figs. 8 and 9. In Fig. 8, the $x$-axis indicates the maximum number of routes per train and the $y$-axis shows the average delay in seconds for the four control areas. In Fig. 9, the $x$-axis indicates the maximum number of routes per train and the $y$-axis shows the number of instances solved to the optimum within the available computational time, again for the four control areas. These figures show the following results.

For the Rouen instances (see Figs. 8 and 9, bottom right) the higher the maximum number of routes per train is, the more the average delay found by RECIFE-MILP increases. Instead, the average delay found by 3BA remains stable and tends to slightly decrease. When the maximum number of routes per train is in the interval (Pellegrini et al., 2015; D'Ariano et al., 2007), the two algorithms achieve the same performance. When this value is 20, RECIFE-MILP solves all the instances to the optimum, while 3BA fails to do so in 12 cases. However, for 8 of them, 3BA reaches the optimal solution although it does not manage to prove its optimality. To assess the statistical significance of the difference between the two algorithms, we perform the Wilcoxon rank-sum test with a confidence level of 0.95. This test does not show any significant difference between the algorithms run on Rouen, when the maximum number of routes per train is in [1, 40]. Nevertheless, for a maximum number of routes per train going from 60 to the extreme value of 384, the difference is statistically significant in favor of 3BA. However, different observations are suggested for the three other control areas. In Gonesse (see Figs. 8 and 9, top left), the experiments do not allow to remark any significant difference when the maximum number of routes per train is smaller than or equal to 2. From 3, this difference is statistically significant in favor of RECIFE-MILP. Moreover, in terms of the number of optima proven, RECIFE-MILP solves all the instances to the optimum, while the number of instances solved to the optimum by 3BA tends to decrease when the maximum number of routes per train increases. In the Lille instances (see Figs. 8 and 9, top right), the bad performance of 3BA is clearly revealed compared to RECIFE-MILP. Even if 3BA manages to solve some instances to the optimum beyond the value of one route per train, RECIFE-MILP outperforms significantly 3BA from two routes per train. In St. Lazare (see Figs. 8 and 9, bottom left), the performance of 3BA is very poor compared to RECIFE-MILP. 3BA fails to prove the optimality for all instances when the maximum number of routes per train exceeds 1. From 2, the difference is statistically significant in favor of RECIFE-MILP.

As a result, we cannot state that the difference of performance between the two algorithms is due to the maximum number of alternative routes available per train. To further investigate the reason of this difference, we study the performance depending to the total number of $x$ and $y$-variables separately. The graphical analysis of this study is done in Figs. 10 and 11.

In Fig. 10, we analyze the behavior of the two algorithms on the four control areas in terms of the number of instances solved to the optimum ($y$-axis) depending the total number of $x$-variables ($x$-axis). For Gonesse, we observe that RECIFE-MILP solves all instances to the optimum, while 3BA fails to do so as soon as more than one route is considered per train (all the points following the first one).
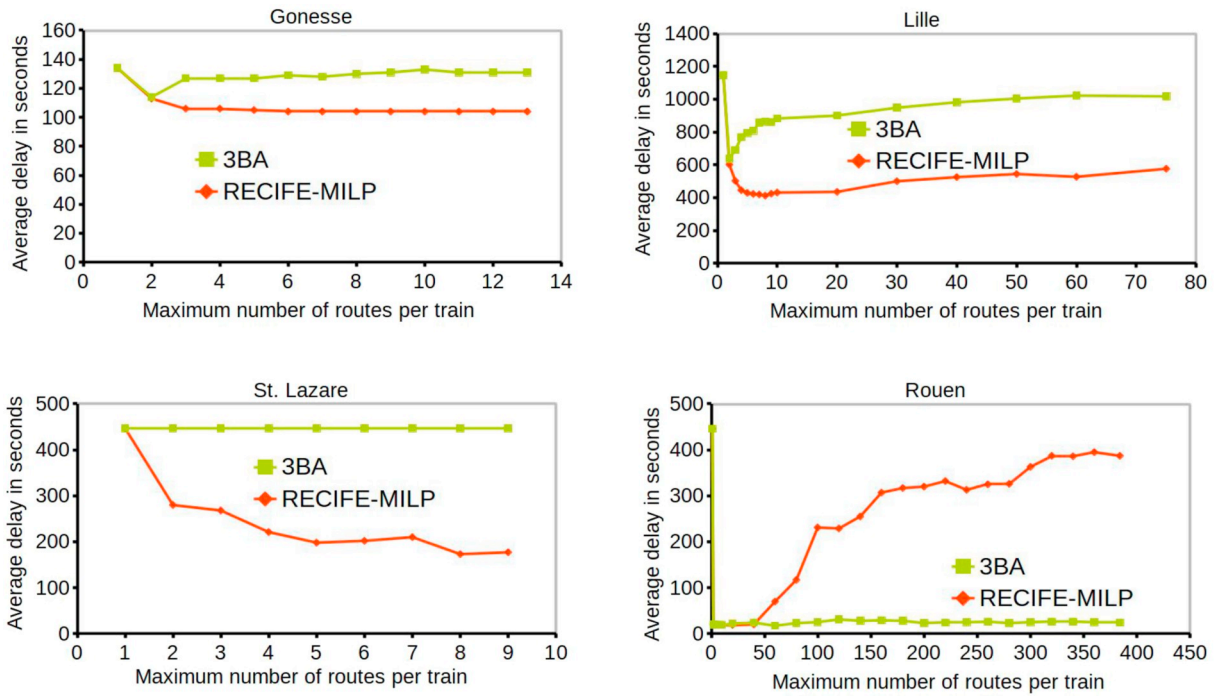
**Fig. 8.** Comparison of the average delay achieved by RECIFE-MILP and 3BA on the four control areas depending the maximum number of routes per train.
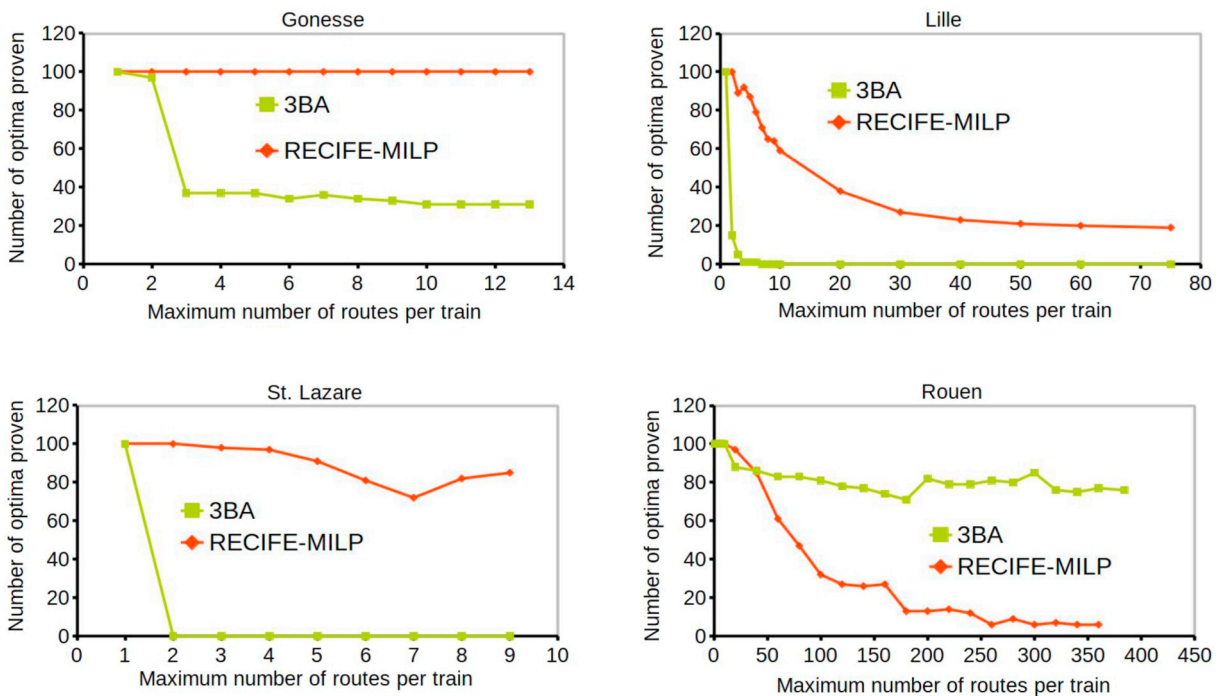


**Fig. 9.** Comparison of the number of instances solved to the optimum by RECIFE-MILP and 3BA on the four control areas depending the maximum number of routes per train.
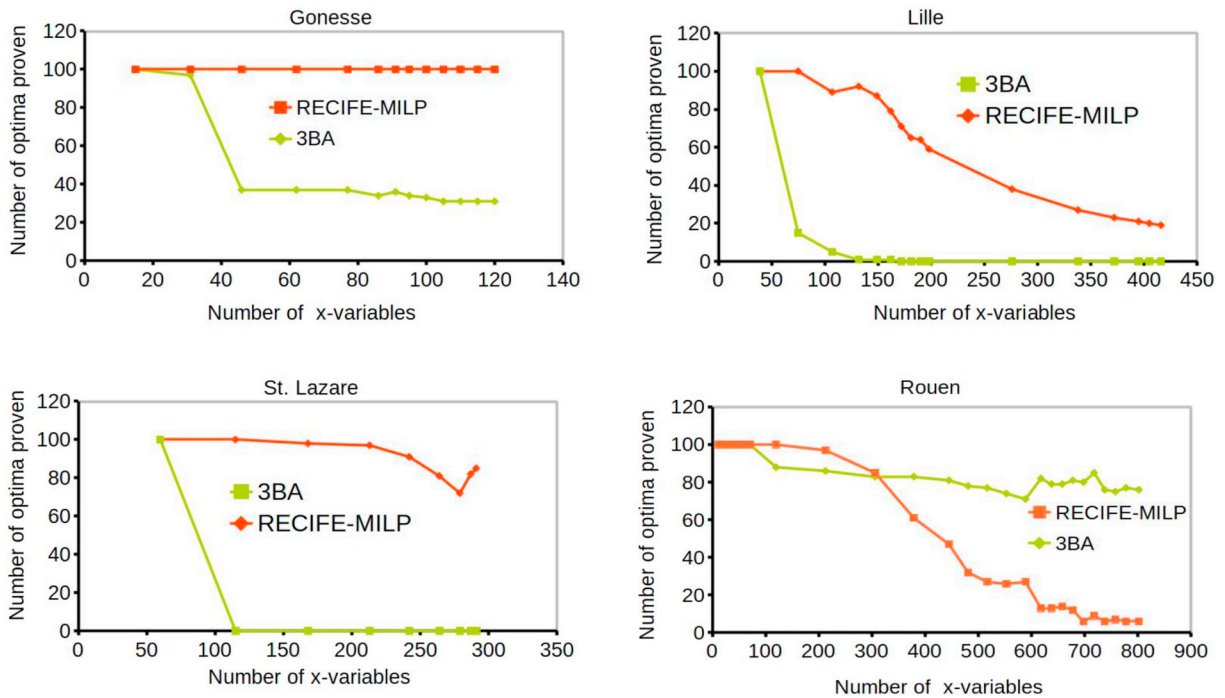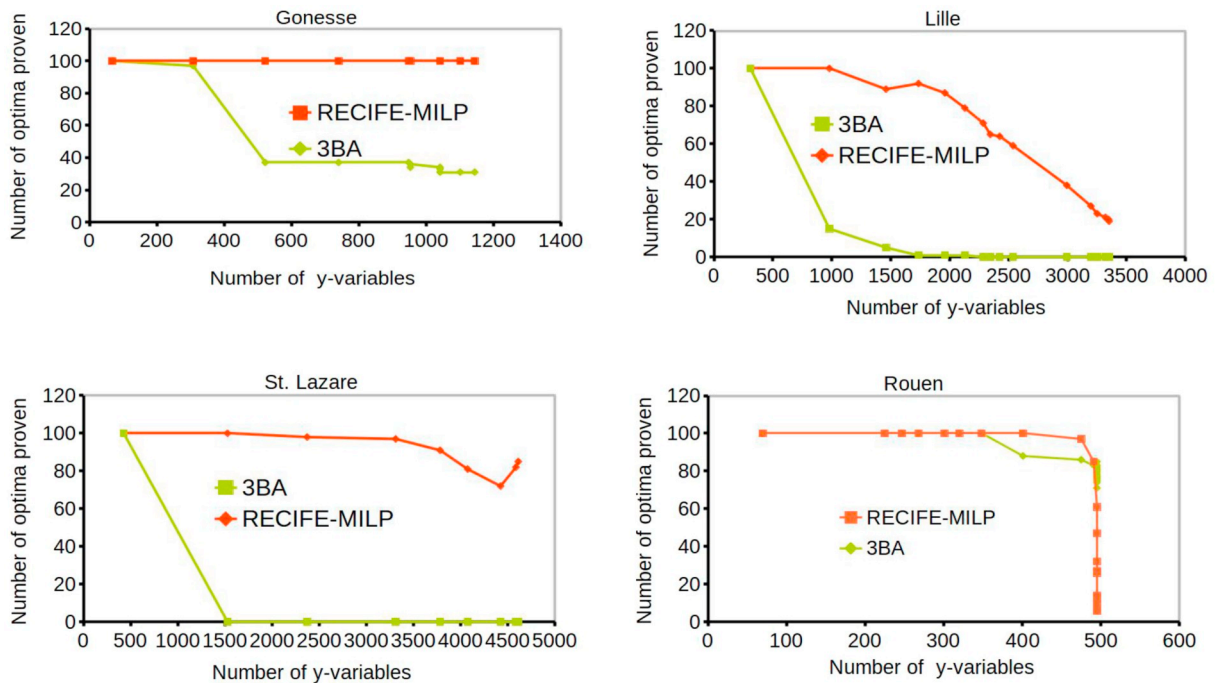
**Fig. 10.** Comparison of the number of instances solved to the optimum by RECIFE-MILP and 3BA on the four control areas depending the number of *x*-variables.



**Fig. 11.** Comparison of the number of instances solved to the optimum by RECIFE-MILP and 3BA on the four control areas depending the number of *y*-variables.

Similarly, we can remark that the performance of 3BA is very bad for the St. Lazare and Lille instances. In St. Lazare, 3BA fails to solve all instances to the optimum even with small numbers of *x*-variables. In the Lille instances, we have a similar behavior of 3BA. As soon as the total number of *x*-variables increases, the number of instances solved by 3BA decreases drastically. When the total number of *x*-

variables reaches 172, 3BA fails to solve all instances. The performance of RECIFE-MILP in the two control areas (St. Lazare and Lille) is also worse than the one in Gonesse. However, although the number of instances solved by RECIFE-MILP tends to decrease, RECIFE-MILP manages to do better than 3BA in the St. Lazare and Lille instances. Instead, for Rouen, when the total number of *x*-variables is in the interval [12, 71], the number of instances solved to the optimum within the available computational time by the two algorithms is the same. When the value is 120 (the first point where the lines get separated in Fig. 10, bottom right), the number of instances solved to the optimum within the available computational time decreases for 3BA and remains constant for RECIFE-MILP. However, when the total number of *x*-variables increases, RECIFE-MILP suffers from this augmentation and its number of instances solved to the optimum tends to decrease, while these figures for 3BA tends to stabilize. The observation that we can draw based on Fig. 10 is that, when an instance of the rtRTMP contains "many" alternative train routes, meaning "many" routes are available for many trains, 3BA seems to be the best option. The results in the different control areas appear coherent in this sense: in Gonesse, where few *x*-variables are present even when all alternative routes are considered for each train, RECIFE-MILP is consistently better than 3BA. The same holds when not all routes are considered for Lille, St. Lazare and Rouen. However, as the number of *x*-variables increases and gets around a few hundreds, the difference in performance becomes smaller in these three control areas: the performance of RECIFE-MILP starts degrading and getting closer to the one of 3BA. In Rouen, where 3BA is capable of solving to optimality a large number of instances, the latter eventually outperforms RECIFE-MILP when the number of *x*-variables becomes very large.

This observation does not fully explain the different performance. Specifically, it does not explain why 3BA is really not well behaving in Lille and St. Lazare. To shed some light on this behavior, we consider the total number of *y*-variables. Similarly to what we just did for *x*-variables, in Fig. 11, we analyze the behavior of the two algorithms on the four control areas in terms of the number of instances solved to the optimum (*y*-axis) depending the total number of *y*-variables (*x*-axis). Note that the number of *y*-variables is constant for Rouen throughout the increase of the number of routes available per train, when this number is above 80: the last points in Fig. 11 bottom right have the same abscissa. This is due to the characteristics of the additional routes available for the trains: they are new combinations of parts of routes which are already covered by existing *y*-variables, since they are present in other trains routes. Instead, in the other three control areas, the number of *y*-variables keeps increasing as the number of alternative routes of each train grows. For Rouen, we observe that for a number of *y*-variables in the interval [70, 348], the two algorithms are successful for all instances. When this value is 401, the number of instances solved by 3BA decreases, while for RECIFE-MILP it remains stable. When the total number of *y*-variables reaches 495, the number of instances solved by 3BA tends to stabilize, while this value for RECIFE-MILP strongly decreases following the increase of the number of routes per train. Instead, for Gonesse (Fig. 11 top left), as soon as more than one route is available for each train (second point in the series), 3BA suffers from the noticeable augmentation of *y*-variables and the number of instances it solves to the optimum decreases, while for RECIFE-MILP it remains stable. Similarly, in the St. Lazare and Lille instances, as the total number of *y*-variables increases RECIFE-MILP does better than 3BA. In Lille, although the number of instances solved to the optimum by the two algorithms decreases when the total number of *y*-variables increases, 3BA suffers from this augmentation more than RECIFE-MILP. Worse still, in the St. Lazare instances, we can remark that 3BA fails to solve all instances to the optimum when the number of routes available per train exceeds 1. An explanation to this bad result achieved by 3BA in St. Lazare can be the immediate predominance of the scheduling decisions (4611) on the routing ones (291).

The conclusion which we draw from these analyses is that, when an instance of the rtRTMP contains "many" *x*-variables and "few" *y*-variables as in the case of Rouen with many alternative routes per train, 3BA achieves a better performance than RECIFE-MILP. Instead, when the number of *y*-variables is "large" and the number of *x*-ones is "small", as in Gonesse, Lille and St. Lazare, RECIFE-MILP outperforms 3BA. These results suggest that, depending on the characteristics of the rtRTMP instances, either 3BA or RECIFE-MILP can be the best option. Specifically, when a rtRTMP instance contains "many" train routes that do not generate "many" rescheduling variables 3BA proves to be efficient. Instead, disregard the number of train routes, if these routes generates "many" rescheduling variables, then RECIFE-MILP seems to be the best.

The fact that 3BA outperforms RECIFE-MILP when there are "many" routing decisions (i.e., *x*-variables) and "few" scheduling ones (i.e., *y*-variables) may be due to several reasons.

One possible reason is related to the computational time necessary to solve the rescheduling problem (i.e., the MILSP) in 3BA. When the MILSP contains "many" *y*-variables this computational time may increase. As a result, 3BA generates less Benders cuts than when the MILSP is easy to solve. We remark that for Gonesse, Lille and St. Lazare, where the MILSP contains "many" *y*-variables "few" Benders cuts are added to the UMP: for example, when all the routes of the original instances are available for the trains, the average computational time per cut added is 1.38, 1.40 and 2.42 s for Gonesse, Lille and St. Lazare, respectively. This corresponds to the generation of 130, 129 and 75 cuts in average for the three control areas in the 3 min computational time available. Instead, for Rouen, where the MILSP contains "few" *y*-variables, "many" Benders cuts are added to the UMP: the average computational time per cut added is 0.70 s, i.e., in average 257 cuts are generated. Indeed, the quality of the search process being equal, the more solutions are explored, and hence the more cuts are generated, the better the solution quality is.

Moreover, we believe that another reason for the different performance is related to the characteristics of the control area considered. In particular, it is related to the fact that different control areas may feature different route inter-dependencies. Let us call two routes interdependent if two trains using them incur in strong potential conflicts, and independent otherwise. A proxy for routes interdependency is the number of *y*-variables which derive from their use: the more the variables, the higher the interdependency.[1] Although routes interdependency is difficult to quantify, intuitively it can be observed that Rouen, with its line structure, allows the

---

[1] Remark that this is only a proxy since what really matters for the reasoning which follows is the presence of potential conflicts, which also depend on the trains timing. However, we think the proxy is meaningful here.

definition of more independent routes than Gonesse, Lille and St. Lazare, that are junctions rich of switches and bidirectional tracks. For Rouen, given a set of trains, a few of which are perturbed, it is often possible to identify an allocation of routes such that the perturbed trains seldom cross the others. Depending on the fact that these routes are chosen or not, the total delay and hence the objective function value of the LSP will change quite a lot. For Gonesse, for example, this is still possible if the number of trains is low, but it becomes very unlikely for large train sets. As a consequence, the objective function value of the LSP corresponding to different route allocations will often be similar for this control area. This is indeed what we observe throughout 3BA runs. Getting now to the different behavior of 3BA on the four control areas, we think that, in general, largely different LSP (and hence DLSP) objective function values will imply largely different coefficients for the *x*-variables involved. In turn, this will imply that different route assignments, when evaluated in the UMP already including some cuts, will have quite different impacts on the value of variable *z*, and hence on the UMP objective function. Thus, the quality of different route assignments may become visible already after a few cut generations, and this may indeed improve the performance of the solution process. Hence for Rouen, where routes are more independent, 3BA soon manages to make wise route choices and reach good performance, whereas for Gonesse, Lille and St. Lazare, where routes are strongly interdependent, 3BA struggles to identify promising assignments and behaves quite poorly.

## 8. Conclusions and future works

In this paper, we aimed at improving the performance of a state-of-the-art algorithm for the rtRTMP when tackling large problem instances. Specifically, we proposed 3BA. It is an algorithm based on Benders-like decomposition of the RECIFE-MILP formulation into three problems: the UMP, the MILSP and the LSP. The UMP contains only the binary re-routing variables. Given the re-routing decisions made in the UMP, the MILSP contains the binary rescheduling variables and the continuous re-timing variables. Based on the optimal scheduling decisions obtained by solving the MILSP, the dual of the LSP defines the cuts to be added to the UMP. This reformulation allows the generation of Benders cuts without the big-M parameter and the avoidance of the exploration of rescheduling infeasible or sub-optimal solutions, which characterize the standard Benders decomposition of RECIFE-MILP.

We tested this algorithm on four French control areas with rather different characteristics. In the experimental analysis, we observed that for the four control areas tackled, 3BA outperforms CBA on the three ones and have the same performance as CBA on one control area. 3BA also clearly outperforms RECIFE-MILP for one control area, while the opposite holds for the three others. We conjecture that the different performance on the four control areas is due to the characteristics of the rtRTMP instances to be tackled: we think that when a rtRTMP instance contains "many" alternative routes for the trains that do not require to make "many" rescheduling decisions 3BA is the best option to achieve high quality results, RECIFE-MILP being preferable otherwise. This is probably due, at least in part, to the time necessary to solve the MILSP when many rescheduling decisions are to be made.

In future research, we will study possible ways for reducing this time, for example through the introduction of valid inequalities.

## Acknowledgment

## References

Acuna-Agost, R., Michelon, P., Feillet, D., Gueye, S., 2011. A MIP-based local search method for the railway rescheduling problem. Networks 57 (1), 69–86.

Benders, J., 1962. Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4, 238–252.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., Wagenaar, J., 2014. An overview of recovery models and algorithms for real-time railway rescheduling. Transp. Res. Part B Methodol. 63, 15–37.

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

Caimi, G., Fuchsberger, M., Laumanns, M., Lüthi, M., 2012. A model predictive control approach for discrete-time rescheduling in complex central railway station approach. Comput. Oper. Res. 39, 2578–2593.

Corman, F., Meng, L., 2015. A review of online dynamic models and algorithms for railway traffic management, intelligent transportation systems. IEEE Trans. 16 (3), 1274–1284.

Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2010. A tabu search algorithm for rerouting trains during rail operations. Transp. Res. Part B 44, 175–192.

Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2012. Optimal inter-area coordination of train rescheduling decisions. Transp. Res. Part E 48, 71–88.

Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2014. Dispatching and coordination in multi-area railway traffic management. Comput. Oper. Res. 44, 146–160.

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

D'Ariano, A., Pacciarelli, D., Pranzo, M., 2007. A branch and bound algorithm for scheduling trains in a railway network. Eur. J. Oper. Res. 183, 643–657.

Fang, W., Yang, S., Yao, X., 2015. A survey on problem models and solution approaches to rescheduling in railway networks, Intelligent Transportation Systems, IEEE Transactions onto appear.

IBM, 2017. Ibm ilog cplex concert technology for c++. https://developer.ibm.com/docloud/blog/2016/11/11/whats-in-cos-12-7/.

Kecman, P., Corman, F., D'Ariano, A., Goverde, R., 2013. Rescheduling models for railway traffic management in large-scale networks. Pub. Transp. 5 (1–2), 95–123.

Keita, K., 2017. Benders Decomposition for the Real-Time Railway Traffic Management Problem (Ph.D.thesis, Centrale Lille).

Lamorgese, L., Mannino, C., 2015. An exact decomposition approach for the real-time train dispatching problem. Oper. Res. 63 (1), 48–64.

Lamorgese, L., Mannino, C., 2016. A non-compact formulation for job-shop scheduling problem in transportation. Report F27646, April SINTEF Tech.

Lamorgese, L., Mannino, C., Piacentini, M., 2016. Optimal train dispatching by benders'-like reformulation. Transp. Sci. 50 (3), 910–925.

Lusby, R., Larsen, J., Ehrgott, M., Ryan, D., 2012. A set packing inspired method for real-time junction train routing. Comput. Oper. Res. 40 (3), 713–724.

Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an n-track network: a model reformulation with network-based cumulative flow variables. Transp. Res. Part B Methodol. 67, 208–234.

Pachl, J., 2009. Railway operations processes. In: Theeg, G., Vlasenko, S. (Eds.), Railway Signaling and Interlocking, vol. 25. International Compendium, pp. 39–60.

Pellegrini, P., Marlière, G., Rodriguez, J., 2014. Optimal train routing and scheduling for managing traffic perturbations in complex junctions. Transp. Res. Part B 59, 58–80.

Pellegrini, P., Marlière, G., Pesenti, R., Rodriguez, J., 2015. Recife-milp: an effective milp-based heuristic for the real-time Railway Traffic Management Problem, intelligent transportation systems. IEEE Trans. 16 (5), 2609–2619.

Pellegrini, P., Marlière, G., Rodriguez, J., 2016. A detailed analysis of the actual impact of real-time railway traffic management optimization. J. Rail Transp. Plan. Manag. 6 (5), 13–31.

Quaglietta, E., Pellegrini, P., Goverde, R., Albrecht, T., Jaekel, B., Marlière, G., Rodriguez, J., Dollevoet, T., Ambrogio, B., Carcasole, D., Giaroli, M., Nicholson, G., 2016. The on-time real-time railway traffic management framework: a proof-of-concept using a scalable standardised data communication architecture. Transp. Res. Part C 3 (4), 137–149.

Rodriguez, J., 2007. A constraint programming model for real-time train scheduling at junctions. Transp. Res. Part B 41, 231–245.

Sahin, I., 1999. Railway traffic control and train scheduling based on inter-train conflict management. Transp. Res. Part B 33, 511–534.

Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., Pacciarelli, D., 2016. Ant colony optimization for the real-time train routing selection problem. Transp. Res. Part B 85, 89–108.

Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., Pacciarelli, D., 2017. On the tactical and operational train routing selection problem. Transp. Res. Part C 76, 1–15.

Samà, M., Meloni, C., D'Ariano, A., Corman, F., 2015. A multi-criteria decision support methodology for real-time train scheduling. J. Rail Transp. Plan. Manag. Appear.

Törnquist Krasemann, J., 2015. Computational decision-support for railway traffic management and associated configuration challenges: an experimental study. J. Rail Transp. Plan. Manag. Appear.

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

# Update

## Journal of Rail Transport Planning & Management

Contents lists available at ScienceDirect

# Journal of Rail Transport Planning & Management

journal homepage: http://www.elsevier.com/locate/jrtpm

# Erratum regarding previously published articles

The Declaration of Competing Interest statements were not included in the published version of articles that appeared in Volume 2 of Journal of Rail Transport Planning & Management. For the below articles, the authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Coordinated railway traffic rescheduling with the Resource Conflict Graph model [J. Rail Transport. Plann. Manag., 2019; 15C]. The authors were contacted after publication to request a Declaration of Interest statement.

[J. Rail Transport. Plann. Manag., 2019; 11C] The authors were contacted after publication to request a Declaration of Interest statement.

Modelling passenger interaction process (PIP) framework using ISM and MICMAC approach [J. Rail Transport. Plann. Manag., 2019; 14C] The authors were contacted after publication to request a Declaration of Interest statement.

Balancing organizational and academic research: Investigating train traffic Controller's geographical workspace design and team situation awareness using gaming simulations [J. Rail Transport. Plann. Manag., 2019; 10C: 34-45].

The indirect costs assessment of railway incidents and their relationship to human error - The case of Signals Passed at Danger [J. Rail Transport. Plann. Manag., 2019; 9C: 34-45].

Safety risks in rail stations: An interactive approach [J. Rail Transport. Plann. Manag., 2019; 11C] The authors were contacted after publication to request a Declaration of Interest statement..

Secure rail station–Research on the effect of security checks on passenger flow [J. Rail Transport. Plann. Manag., 2019; 10C: 9-22].

Impact of on-track competition on public finances–The case of the Czech Republic [J. Rail Transport. Plann. Manag., 2019; 12C] The authors were contacted after publication to request a Declaration of Interest statement..

System train paths as key to efficient infrastructure usage for on-track competition in ITF-regimes [J. Rail Transport. Plann. Manag., 2019; 12C] The authors were contacted after publication to request a Declaration of Interest statement..

Optimization of handouts for rolling stock rotations [J. Rail Transport. Plann. Manag., 2019; 10C: 1-8].

Comparing train driving strategies on multiple key performance indicators [J. Rail Transport. Plann. Manag., 2019; 13C] The authors were contacted after publication to request a Declaration of Interest statement.

A practical approach to revenue management in passenger train services: A case study of the Indonesian railways Argo Parahyangan [J. Rail Transport. Plann. Manag., 2019; 13C] The authors were contacted after publication to request a Declaration of Interest statement.

A three-step Benders decomposition for the real-time Railway Traffic Management Problem [J. Rail Transport. Plann. Manag., 2019; 13C] The authors were contacted after publication to request a Declaration of Interest statement.

Improvement of timetable robustness by analysis of drivers' operation based on decision trees [J. Rail Transport. Plann. Manag., 2019; 9C] The authors were contacted after publication to request a Declaration of Interest statement.

A methodology for the design of sections block length on ETCS L2 railway networks [J. Rail Transport. Plann. Manag., 2019; 13C] The authors were contacted after publication to request a Declaration of Interest statement.

Harvesting railway passenger opinions on multi themes by using social graph clustering [J. Rail Transport. Plann. Manag., 2019;

13C] The authors were contacted after publication to request a Declaration of Interest statement.

A matheuristic method for planning railway freight transportation with hazardous materials [J. Rail Transport. Plann. Manag., 2019; 10C] The authors were contacted after publication to request a Declaration of Interest statement.

Does the incumbent have an advantage in open access passenger rail competition? A case study on the Prague–Brno line [J. Rail Transport. Plann. Manag., 2019; 12C] The authors were contacted after publication to request a Declaration of Interest statement.

A collection of aspects why optimization projects for railway companies could risk not to succeed – A multi-perspective approach [J. Rail Transport. Plann. Manag., 2019; 11C] The authors were contacted after publication to request a Declaration of Interest statement.