



HAL
open science

RECIFE-MILP: An effective MILP-based heuristic for the real-time railway traffic management problem

Paola Pellegrini, Grégory Marliere, Raffaele Pesenti, Joaquin Rodriguez

► **To cite this version:**

Paola Pellegrini, Grégory Marliere, Raffaele Pesenti, Joaquin Rodriguez. RECIFE-MILP: An effective MILP-based heuristic for the real-time railway traffic management problem. *IEEE Transactions on Intelligent Transportation Systems*, 2015, 16 (5), pp. 2609 - 2619. 10.1109/TITS.2015.2414294 . hal-01471388v2

HAL Id: hal-01471388

<https://hal.science/hal-01471388v2>

Submitted on 7 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RECIFE-MILP: An Effective MILP-Based Heuristic for the Real-Time Railway Traffic Management Problem

Paola Pellegrini, Grégory Marlière, Raffaele Pesenti, and Joaquin Rodriguez

Abstract—The real-time railway traffic management problem consists of selecting appropriate train routes and schedules for minimizing the propagation of delay in case of traffic perturbation. In this paper, we tackle this problem by introducing RECIFE-MILP, a heuristic algorithm based on a mixed-integer linear programming model. RECIFE-MILP uses a model that extends one we previously proposed by including additional elements characterizing railway reality. In addition, it implements performance boosting methods selected among several ones through an algorithm configuration tool. We present a thorough experimental analysis that shows that the performances of RECIFE-MILP are better than the ones of the currently implemented traffic management strategy. RECIFE-MILP often finds the optimal solution to instances within the short computation time available in real-time applications. Moreover, RECIFE-MILP is robust to its configuration if an appropriate selection of the combination of boosting methods is performed.

Index Terms—Real-time railway traffic management problem, mixed-integer linear programming, algorithm configuration, performance boosting.

I. INTRODUCTION

WHEN defining the train timetables, railway planners need to exploit all the network capacity during the peak periods. A consequence of this need is that unexpected events perturbing the operations (e.g., the late arrival of a driver) may penalize the traffic regularity due to a possibly remarkable delay propagation. To minimize the delay propagation, dispatchers may decide to partially reroute trains and handle train schedules in strategic locations within a specific traffic control area: they tackle the real-time Railway Traffic Management Problem (rtRTMP) [16].

The literature proposes several solution approaches to the rtRTMP [2]. Among them, we have presented a mixed-integer linear programming (MILP) formulation [16]: it allows solving rtRTMP instances to optimality considering all the alternative routes physically available in the infrastructure of interest. In addition, it also allows modeling the route-lock sectional-

release interlocking system [20]. With few exceptions [18], [21], the other algorithms typically propose heuristic solutions that consider a limited number of alternative routes: they are either selected during the solution process [1], [5], [7], [8], or arbitrarily set *a priori* [3], [4]. This is done both in the algorithms modeling train-speed-variation dynamics at different extents of approximation [13], and in those neglecting it [6], [16]. The formulation in [16] pays in terms of computation time its ability of tackling very numerous alternative routes and of proving the optimality of a solution. A slightly different research stream focuses on the use of simulation tools for assessing the impact of railway management strategies [22], [23].

This paper extends the model in [16]. Moreover, it proposes a heuristic algorithm with the following features:

- 1) it can be applied in a real-time context for managing traffic in complex infrastructures, achieving high quality performance;
- 2) it is robust to the type of infrastructure considered, i.e., it can be effectively used on control areas with very different characteristics;
- 3) it is robust with respect to its own configuration, i.e., the same configuration can be effectively used throughout the control area.

To this end, we first extend the MILP formulation of [16] to describe the characteristics of the infrastructures in deeper detail. Second, we boost the solution algorithm: we develop valid inequalities and algorithmic expedients (some of which were briefly introduced in [15], [17]) and we select the most appropriate ones thanks to a state-of-the-art algorithm-configuration procedure. Finally, we set a time limit so that when this limit is reached, the algorithm returns the best solution it has found so far. Note that the search process is started considering a subregion of the feasible space, where at least one feasible solution is always found very quickly. This algorithm is named RECIFE-MILP. It has been developed as part of the decision support tool named RECIFE (REcherche sur la Capacité des Infrastructures FERroviaires) [19].

We assess the performance of RECIFE-MILP in a thorough experimental analysis based on three French traffic control areas with different characteristics: the Pierrefitte–Gonesse junction; the Lille–Flandres station; the Rouen–Rive–Droite control area. In the literature, very few analysis on the rtRTMP on different control areas are made at the level of detail of the rail equipment considered here. To the best of our knowledge, this has previously been done only in [3]. The previous studies

Manuscript received September 16, 2014; revised January 8, 2015 and March 2, 2015; accepted March 12, 2015. The Associate Editor for this paper was F.-Y. Wang.

P. Pellegrini is with Univ. Lille Nord de France, F-59000 Lille, IFSTTAR, COSYS, LEOST, F-59650 Villeneuve d'Ascq, France.

G. Marlière and J. Rodriguez are with Univ. Lille Nord de France, F-59000 Lille, IFSTTAR, COSYS, ESTAS, F-59650 Villeneuve d'Ascq, France.

R. Pesenti is with Ca' Foscari University, 30121 Venice, Italy.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2015.2414294

focusing on more than one control area [1] introduce some approximations in the representation, concerning for example the number of tracks along a line sections. With the three precisely represented control areas, we cover most of the actual characteristics of the real case-studies and show that RECIFE-MILP is capable of dealing with all the relevant aspects of the railway traffic. The three control areas have previously been object of studies in the context of research projects performed in collaboration with the French infrastructure manager Réseau Ferré de France (RFF) and the French main railway undertaking Société Nationale des Chemins de Fer (SNCF): the Nord Pas De Calais region RECIFE project for the Lille–Flandres station and the Pierrefitte–Gonesse junction, and the PREDIT project SIGIFret (SIMulation d’une Gestionne Innovante des circulations Fret) for the Rouen–Rive–Droite control area.

In summary, the research contribution of this paper is three-fold. First, we broaden the model in [16] to describe in deep detail the element which condition the operational decisions in railway traffic. Second, we provide a solution algorithm which can be applied in a real-time context. In particular, we propose and assess several methods for boosting the algorithm performance. Third, we evaluate both the model and the algorithm robustness by testing their real-time applicability on three control areas with very different characteristics.

The rest of the paper is organized as follows. Section II describes the MILP formulation. Section III presents the valid inequalities and the algorithmic expedients proposed. Section IV details the algorithm-configuration procedure used in the experimental analysis. Section V discusses the experimental setup and the results obtained for each control area. Finally, Section VI concludes the paper.

II. MODEL FORMULATION

In this section, we detail the MILP formulation used and we point out where and why we extend the one in [16].

The infrastructure is modeled considering *track-circuits*, i.e., track segments on which the presence of a train is automatically detected. In addition to the existing track-circuits, we introduce two dummy ones: tc_0 and tc_∞ . They represent the entry and the exit locations of the infrastructure considered, respectively. Sequences of track-circuits are grouped into *block sections*, whose access 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*. We name *utilization* the sum of *reservation* and *occupation* time. If a train starts its *trip* at null speed, in the model its occupation of the first track-circuit is accounted only from the time at which it starts moving. Its staying still on the track-circuit before that time is represented through reservation. The complete sequence of track-circuits traversed by a train during its trip is named *route*. Differently from [16], we consider routes which may include intermediate stops. In the model, a sequence of track-circuits used by both direct and non-direct trains defines two

routes. The running times for the route with intermediate stops include the appropriate deceleration and acceleration times and no dwell time. As explained in the following, the respect of the minimum dwell time is assured through the model constraints.

In the MILP formulation, we use the following notation:

T	set of trains;
w_t	weight associated to train t 's delay;
ty_t	type corresponding to train t (train characteristics);
$init_t, exit_t$	earliest time at which train t can be operated and earliest time at which it can reach its destination given $init_t$, the route assigned in the timetable and the intermediate stops;
$i(t', t)$	indicator function: 1 if t' and t use the same rolling stock and t results from the turnaround, join or split of t' , 0 otherwise;
ms	minimum separation between the arrival and the departure of two trains using the same rolling stock;
R_t, TC_t	set of routes and track-circuits available for train t ;
TC^r	set of track-circuits composing route r ;
$OTC_{ty,r,tc}$	set of track-circuits occupied by a train t of type ty along r if t 's head is at the end of tc (\emptyset if ty is shorter than tc);
$TC(tc, tc', r)$	set of track-circuits between tc and tc' along r ;
$p_{r,tc}, s_{r,tc}$	track-circuits preceding and following tc along r ;
$rt_{ty,r,tc}, ct_{ty,r,tc}$	running and clearing time of tc along r for a train of type ty ;
$ref_{r,tc}$	reference track-circuit for the reservation of tc along r , depending on block-sections structure;
$e(tc, r)$	indicator function: 1 if track-circuit tc belongs to either the first or the last block section of r , 0 otherwise;
$bs_{r,tc}$	block section including track-circuit tc along route r ;
for_{bs}, rel_{bs}	formation and release time for block section bs ;
$S_t, TCS_{t,s}$	set of stations where t has a scheduled stop and set of track-circuits that can be used by t for stopping at s ;
$dw_{t,s}, a_{t,s}, d_{t,s}$	minimum dwell time, scheduled arrival and scheduled departure times for train t at station s ;
M	large constant.

The formulation uses non-negative **continuous variables**: for all triplets of $t \in T$, $r \in R_t$ and $tc \in TC^r$:

$ot_{r,tc}$:	time in which t starts the occupation of tc along r ,
$lt_{r,tc}$:	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 $tc \in TC_t$:

$sU_{t,tc}, eU_{t,tc}$: time in which t starts and ends tc utilization;

for all $t \in T$:

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

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 t is smaller than the index t' , and $tc \in TC_t \cap TC_{t'}$

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

The objective is the minimization of the total weighted delays suffered by trains at their exit from the infrastructure

$$\min \sum_{t \in T} w_t D_t. \quad (1)$$

Such an objective, differently from those considered in [16], is able to capture different train priorities.

The MILP formulation includes constraints that impose the following conditions. A train t cannot be operated earlier than $init_t$ (2). The start time of track-circuit occupation along a route is zero if the route itself is not used (3). 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 (4). A train t with a scheduled stop at station s and using route r does not enter the track-circuit following tc before the scheduled departure time from s if tc is in $TCS_{t,s}$ (5), and it has a longer stay in tc of at least $dw_{t,s}$ (6). A train t must use exactly one route (7). 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 (8). A time ms must separate the arrival and departure of trains using the same rolling stock (9). If trains t' and t use the same rolling stock and t results from t' , the track-circuit tc where the turnaround, join or split takes place must be utilized for the whole time between t' 's arrival and t 's departure. Thus, tc starts being reserved by t at the latest when t' ends its utilization (10). Here, the inequality must be imposed since, in case of a join, two trains arrive and are connected to become a single departing one. The utilization of the departing train must then immediately follow the utilization of the first train arriving, being strictly smaller than the one of the second train. Moreover, a train's utilization of a track-circuit starts as soon as the train starts occupying the track-circuit $ref_{r,tc}$ along one of the routes including it, minus the formation time (11). Constraints (11) are imposed as inequalities (\leq) when they concern a track-circuit of the first block sections of the route ($ref_{r,tc} = s_{r,tc_0}$) and the train t results from the turnaround, join or split of one or more other trains. This is a consequence of the need of keeping platforms utilized. Indeed, if t results from t' , Constraints (10) ensure that the track-circuit where the turnaround takes place starts being

reserved by t as soon as t' arrives. However, t needs to wait at least for a time ms before departing. The occupation of the track-circuit by t is however starting from its actual departure, for guaranteeing the coherence of the occupation variables and the running time (Constraints (4)). Hence, t 's reservation starts much earlier than its occupation. Furthermore, the utilization of a track-circuit lasts till the train utilizes it along any route, plus the formation and the release time (12). Here $ul_{t,r,tc}$ is the total utilization time. It includes: the running time of all track-circuits between $ref_{r,tc}$ and tc , the longer stay of the train's head on each of these track-circuits and the clearing time of tc . Moreover, it includes the longer stay on all track-circuits tc' such that $tc \in OTC_{ty_{t,r,tc'}}$. As mentioned in the description of the notation, if the head of the train is on one of these track-circuits, then its tail has not yet exited tc : the train is longer than tc' , or of the sequence of track-circuits between tc and tc' . Hence, if the train suffers a longer stay when its head is on one of these track-circuits, such a longer stay must be counted in the utilization time of tc . The track-circuit utilizations by two trains must not overlap (13), (14).

$$o_{t,r,tc} \geq init_t x_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r \quad (2)$$

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

$$o_{t,r,tc} = o_{t,r,pr,tc} + l_{t,r,pr,tc} + r_{t,r,ty_{t,pr,tc}} x_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r \quad (4)$$

$$o_{t,r,s_r,tc} \geq \sum_{\substack{s \in S_t: \\ tc \in TCS_{t,s} \cap TC^r}} d_{t,s} x_{t,r} \quad \forall t \in T, r \in R_t, tc \in \bigcup_{s \in S_t} TCS_{t,s} \quad (5)$$

$$l_{t,r,s_r,tc} \geq \sum_{\substack{s \in S_t: \\ tc \in TCS_{t,s} \cap TC^r}} dw_{t,s} x_{t,r} \quad \forall t \in T, r \in R_t, tc \in \bigcup_{s \in S_t} TCS_{t,s} \quad (6)$$

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T \quad (7)$$

$$D_t \geq \sum_{r \in R_t} o_{t,r,tc_\infty} - exit_t \quad \forall t \in T \quad (8)$$

$$\sum_{\substack{r \in R_t, tc \in TC^r: \\ pr,tc = tc_0}} o_{t,r,tc} \geq \sum_{\substack{r \in R_{t'}, tc \in TC^r: \\ sr,tc = tc_\infty}} o_{t',r,tc} + (ms + r_{t,r,ty_{t',tc}}) x_{t',r} \quad \forall t, t' \in T : i(t', t) = 1 \quad (9)$$

$$\sum_{\substack{tc \in TC_t: \\ \exists r \in R_t, pr,tc = tc_0}} sU_{t,tc} \leq \sum_{\substack{tc \in TC_{t'}: \\ \exists r \in R_{t'}, sr,tc = tc_\infty}} eU_{t',tc} \quad \forall t, t' \in T : i(t', t) = 1 \quad (10)$$

$$sU_{t,tc} = \sum_{\substack{r \in R_t: \\ tc \in TC^r}} (o_{t,r,ref_{r,tc}} - for_{bs_{r,tc}} x_{t,r}) \quad \forall t \in T, tc \in TC_t : (\exists t' \in T : i(t', t) = 1) \vee (\forall r \in R_t : ref_{r,tc} \neq s_{r,tc_0}) \quad (11)$$

$$eU_{t,tc} = \sum_{\substack{r \in R_t: \\ tc \in TC^r}} o_{t,r,ref_{r,tc}} + (for_{bs_{r,tc}} + rel_{bs_{r,tc}}) x_{t,r} + ul_{t,r,tc} \quad \forall t \in T, tc \in TC_t \quad (12)$$

$$eU_{t,tc} - M(1 - y_{t,t',tc}) \leq sU_{t',tc} \quad \forall t, t' \in T, \text{index } t < \text{index } t', tc \in TC_t \cap TC_{t'} : i(t, t') \sum_{r \in R_t} e(tc, r) = 0 \wedge i(t', t) \sum_{r \in R_{t'}} e(tc, r) = 0 \quad (13)$$

$$eU_{t',tc} - M y_{t,t',tc} \leq sU_{t,tc} \quad \forall t, t' \in T, \text{index } t < \text{index } t', tc \in TC_t \cap TC_{t'} : i(t, t') \sum_{r \in R_t} e(tc, r) = 0 \wedge i(t', t) \sum_{r \in R_{t'}} e(tc, r) = 0. \quad (14)$$

To sum up, the main differences w.r.t. [16] consist in the account for: i) different train priorities (1); ii) intermediate commercial stops along trips (5), (6); iii) punctual lengths of trains and track-circuits (12). Indeed, the consideration of these additional aspects of railway traffic helps in filling the gap between the reality and the model. However, they constitute additional constraints to be satisfied in the solution process.

The difficulty in solving the MILP model is mostly due to the multiplicity of both the alternative routes and the potential conflicts. Intuitively the higher the number of routes and potential conflicts, the higher the number of alternative solutions to evaluate, and among them the number of the ones with an almost equivalent objective function value. The presence of this latter kind of solutions in turn affects the number of subtrees that a branch-and-bound algorithm cannot discard without a full exploration and then the hardness of the optimality proof.

The observation of the topology of an infrastructure may give a hint on the difficulty of the instances. In particular, if a large number of partially overlapping routes exists, then the instances may be hard to solve. For example, in Lille (Section V-C), a lot of crossing and overlapping alternative routes connect lines to platforms: traffic here is likely to be hard to optimize. As another example, in Rouen (Section V-D), the presence of many stations along the line generates a large number of alternative routes thanks to the combination of platform assignments, which might generate instances with many solutions with almost equivalent objective function value.

III. BOOSTING METHODS

In this section we present several boosting methods aiming to improve the solution algorithm performance. Specifically, their aim is, first, to increase the speed of detection of high quality feasible solutions and, second, to decrease the computation time needed for proving optimality. None of these methods is present in the analysis proposed in [16], but a slightly more elementary version of the Initialization one.

Initialization—Provision of an Initial Solution: As in [16], we implement a two-optimization-step cycle. In the first step, the MILP solver optimizes the train scheduling without modifying the routes with respect to the scheduled ones. In the second step, it optimizes also in terms of train routing, using as initial solution the best solution found in the first step. In addition to what was done in [16], here we initialize the whole algorithm by providing as an initial solution to the first step the greedy solution obtained assigning track-circuits to the first train claiming them (first-come-first-served). Since we need an extremely quick approach, we do not implement any sophisticated control on the future implication of track-circuit utilization. If a long enough sequence of block sections must be traversed by two trains in opposite directions, it is possible that each of these trains is allowed to enter the sequence concurrently. In this case, the two trains end up in a deadlock. For such a situation, we just consider that the two trains could not reach their destination, and we do not include them in the initial solution used: the MILP solver starts from a partial solution. Disregarding the optimality proof, we terminate the

first step after 30 seconds 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 rerouting (first optimization step) finds a feasible solution within very few seconds.

Topology—Infrastructure Topology Exploitation: We reduce as much as possible the number of binary y -variables, i.e., the variables appearing in Constraints (13) and (14) that define the precedence relation between couples of trains utilizing the same track-circuit. To this aim, we exploit 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. Consider as an example a bidirectional track without any overpassing area. In a preprocessing phase, for each couple of trains t, t' and on the basis of the physical network topology, we first partition the set of shared track-circuits $TC_t \cap TC_{t'}$ into subsets defined as follows: two track-circuits belong to the same subset \hat{TC} if the same precedence relation between t and t' must hold on both track-circuits. Then, for each \hat{TC} , a single $y_{t,t',\hat{tc}}$ variable is associated to the $tc \in \hat{TC}$ with the lowest index, and we replace $y_{t,t',tc}$ with $y_{t,t',\hat{tc}}$ in Constraints (13) and (14).

M-Decrease—Decrease of the Value of the Large Constant: We exploit the solution obtained in the first optimization step for decreasing the value of M , the large constant used in the formulation. For ensuring the coherence of Constraints (13) and (14), M needs to be at least equal to the latest end of a concerned track-circuit utilization. Let S^{*1} be the optimal solution in the first optimization step and wD^{*1} its total weighted delay. In the second optimization step, all solutions improving over S^{*1} will have an associated total weighted delay not greater than wD^{*1} . Thus, the latest utilization of a track-circuit tc by a train t will be at most equal to the sum of wD^{*1}/w_t and the maximum of the earliest possible exit time from tc along each available route. Formally, let $\bar{u}_{t,tc}$ be the latest time at which t may end the utilization of track-circuit tc during the second optimization step: if no stops are scheduled between the beginning of the route and tc , then

$$\bar{u}_{t,tc} \leq \frac{wD^{*1}}{w_t} + \max_{r \in R_t} \left\{ \begin{aligned} &init_t + \sum_{tc' \in TC(tc_0, tc, r)} rt_{r,ty_t,tc'} \\ &+ ct_{r,ty_t,tc} + rel_{bs_r,tc} \end{aligned} \right\};$$

otherwise

$$\bar{u}_{t,tc} \leq \frac{wD^{*1}}{w_t} + \max_{r \in R_t} \left\{ \begin{aligned} &d_{t,s} + \sum_{tc' \in TC(tc'', tc, r)} rt_{r,ty_t,tc'} \\ &+ ct_{r,ty_t,tc} + rel_{bs_r,tc} \end{aligned} \right\}$$

if the last stop before tc is in tc'' at station s . M can be set equal to the maximum of these quantities across the trains involved in each constraint.

Shift—Backward Shift of Reference Times: We decrease all $init_t$'s, $exit_t$'s, $a_{t,s}$'s and $d_{t,s}$'s of the quantity $\min_{t \in T} \{init_t - \max_{r \in R_t, tc \in TC^r} \{for_{bs_r, tc}\}\}$. Such a backward shift allows the MILP solver to deal with smaller quantities and it amplifies the impact of the M-decrease boosting method.

XY-Relation—Explicit Relation Between x and y -variables: We insert two sets of valid inequalities in the model, aiming to reduce the number of equivalent sub-trees to be explored by the MILP solver. They are based on the observation that, for each couple of trains t and t' and common track-circuit tc , both assignments $y_{t,t',tc} = 1$ and $y_{t,t',tc} = 0$ lead to equal solution values if neither t nor t' use a route passing by tc . To avoid the exploration of both sub-trees, we impose $y_{t,t',tc} = 0$ unless at least one of the trains uses tc :

$$y_{t,t',tc} \leq \sum_{r \in R_t: tc \in TC^r} x_{t,r} + \sum_{r \in R_{t'}: tc \in TC^r} x_{t',r} \quad \forall t, t' \in T, \text{index } t < \text{index } t', tc \in TC_t \cap TC_{t'}. \quad (15)$$

X-Priority—Highest Priority to Route Assignment Variables: We impose high priority to the branching on x -variables during the MILP solution. This method might be particularly useful in combination with the XY-relation one.

Transitivity—Transitivity of the Precedence Relations: We add a set of valid inequalities which impose the transitivity between triplets of y -variables:

$$y_{t,t',tc} + y_{t',t'',tc} \leq 1 + y_{t,t'',tc} \quad \forall t, t', t'' \in T$$

$$\text{index } t < \text{index } t' < \text{index } t'', tc \in TC_t \cap TC_{t'} \cap TC_{t''} \quad (16)$$

$$y_{t,t',tc} + y_{t',t'',tc} \geq y_{t,t'',tc} \quad \forall t, t', t'' \in T$$

$$\text{index } t < \text{index } t' < \text{index } t'', tc \in TC_t \cap TC_{t'} \cap TC_{t''}. \quad (17)$$

In particular, Inequalities (16) impose that if $t \prec t'$ and $t' \prec t''$, then $t \prec t''$. Inequalities (17) impose the opposite relation.

Earliest-Start: Separation of Utilization Start for Couples of Trains: We add a set of valid inequalities which exploit the following observation. For each track-circuit tc certainly used by a couple of trains t and t' , i.e., belonging to all routes of both trains, Constraints (13) and (14) impose a minimum headway between their utilization starts. This time is related to the structure of the block sections along which tc can be used and on the earliest time at which one of the two trains can reach tc . In fact, let $\hat{o}_{t,tc}$ be the earliest time at which t may start the occupation of tc given $init_t$, the scheduled intermediate stops and the shortest available partial route connecting tc_0 to tc . If no stops are scheduled between tc_0 and tc itself then

$$\hat{o}_{t,tc} = \min_{r \in R_t} \left\{ init_t + \sum_{tc' \in TC(tc_0, tc, r) \setminus \{tc\}} rt_{r, ty_t, tc'} \right\}.$$

Otherwise, if the last stop before tc is in tc'' at station s then

$$\hat{o}_{t,tc} = \min_{r \in R_t} \left\{ d_{t,s} + \sum_{tc' \in TC(tc'', tc, r) \setminus \{tc\}} rt_{r, ty_t, tc'} \right\}.$$

In addition, let $\hat{o}_{t,t',tc} = \min\{\hat{o}_{t,tc}, \hat{o}_{t',tc}\}$ and $\hat{U}_{t,tc} = \min_{r \in R_t} ul_{t,r,tc}$, i.e., the minimum utilization time of tc for t along any of its routes. We explicitly impose the minimum separation through the following set of inequalities

$$(sU_{t,tc} - \hat{o}_{t,t',tc})\hat{U}_{t,tc} + (sU_{t',tc} - \hat{o}_{t,t',tc})\hat{U}_{t',tc} \geq \hat{U}_{t,tc}\hat{U}_{t',tc} \quad \forall t, t' \in T, t < t', tc \in TC_t \cap TC_{t'} : tc \in TC^r \quad \forall r \in R_t, tc \in TC^r \quad \forall r \in R_{t'}. \quad (18)$$

To better understand the meaning of (18), consider the following example: $t \prec t'$ on tc ; t is the first one that may arrive at tc given $init_t$ and $init_{t'}$ and the available routes ($\hat{o}_{t,t',tc} = \hat{o}_{t,tc}$); t starts the utilization as soon as it can, i.e., at $\hat{o}_{t,tc}$. Then, $sU_{t,tc} - \hat{o}_{t,t',tc} = 0$ and the inequality states that

$$(sU_{t',tc} - \hat{o}_{t,t',tc})\hat{U}_{t',tc} \geq \hat{U}_{t,tc}\hat{U}_{t',tc} \Rightarrow$$

$$\Rightarrow (sU_{t',tc} - sU_{t,tc})\hat{U}_{t',tc} \geq \hat{U}_{t,tc}\hat{U}_{t',tc}$$

$$\Rightarrow sU_{t',tc} - sU_{t,tc} \geq \hat{U}_{t,tc}$$

i.e., the start of the utilization of t' must follow the one of t of at least the minimal amount of time for which t utilizes tc . This latter condition is fully expressed by Constraints (13) and (14), if M is a large value, only when the y -variables assume an integer value.

LX-Relation—Explicit Relation of Longer Stay and Route: We impose a set of inequalities stating that, if a route r is not used by a train, all its longer stay variables along r are null:

$$l_{t,r,tc} \leq Mx_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r. \quad (19)$$

Distant-Trains—Imposing Implicit Precedence Between Trains: We perform a preprocess of the y -variables between the two optimization steps. Specifically, let S^{*1} be again the optimal solution in the first optimization step, with total weighted delay wD^{*1} . Let $\bar{u}_{t,tc}$ be the latest time at which t may end the utilization of tc during the second optimization step, as defined for the M-decrease method. Then, the following conditions hold in the second optimization step:

$$y_{t,t',tc} = 0 \quad \forall t, t' \in T, \text{index } t < \text{index } t'$$

$$tc \in TC_t \cap TC_{t'}, init_t > \bar{o}_{t',tc}, \bar{\exists} r \in R_t : tc \notin TC^r \quad (20)$$

$$y_{t,t',tc} = 1 \quad \forall t, t' \in T, \text{index } t < \text{index } t'$$

$$tc \in TC_t \cap TC_{t'}, init_{t'} > \bar{u}_{t,tc}, \bar{\exists} r \in R_t : tc \notin TC^r. \quad (21)$$

They state that if tc belongs to all routes of t and t' and t surely reaches tc earlier than the arrival time of t' in the infrastructure ($init_{t'} > \bar{u}_{t,tc}$) then $t \prec t'$.

IV. SMAC

Boosting methods must be used with due care. On the one side, a method may significantly improve the algorithm performance. On the other side, it introduces some computational overhead that in the worst case may outweigh the method advantages. Consequently, given a class of instances, e.g., the ones that may occur at a given station, it is opportune to find the appropriate algorithm configuration. Hereinafter, an algorithm configuration is a set of activated boosting methods proposed in Section III.

Unfortunately, it is hardly possible to compare in a statistically meaningful way all the possible 2^{10} algorithm configurations. Hence, we used an algorithm configuration procedure in the experimental analysis: the Sequential Model-based Algorithm Configuration (SMAC). SMAC is a sequential model-based optimization method [9] which tackles the algorithm configuration problem. Its potential has been assessed in several applications. Given a class of instances Π and a set of boosting methods, SMAC selects the most appropriate configuration on the basis of a randomly selected sample $\tilde{\Pi} \subset \Pi$ of the instances. SMAC works off-line and returns a configuration aimed to be used in all algorithm runs on any instance in Π . SMAC considers the algorithm to be configured as a black-box and it observes the algorithm behavior under different configurations on $\tilde{\Pi}$. It searches for the best configuration by implementing the following iterative heuristic:

- 1) (*Initialization*) The default configuration is run on the instances in $\tilde{\Pi}$ and is defined incumbent configuration;
- 2) (*Iteration*)
 - i) a multi-start local search procedure is used to select a set of new configurations to be assessed (challengers) in the neighborhood of the best ones tested so far;
 - ii) the performance of the challengers is compared against the one of the incumbent through the machine learning tool named racing algorithm [14]. After the assessment, if appropriate, the incumbent is replaced.

The key component of SMAC is step 2.ii. There, SMAC initially predicts each challenger performance. Without the need of actually running the algorithm, the prediction is based on the performance observed for previously run configurations. Then, if the predicted performance is poor, the challenger is rejected. Otherwise, the algorithm is run with the challenger configuration on instances sampled uniformly at random from those on which the incumbent has already run. If at some point the challenger's empirical performance is worse than that of the incumbent configuration, then the challenger is rejected. Differently, the assessment of the challenger stops when the number of runs performed reaches the one of the incumbent. In this case, the incumbent is replaced by the challenger.

The predicted performance and uncertainty of a challenger are measured through a machine learning tool named random forest [10]. The input of this tool is a table which reports the performance of the configurations tested so far. In this table, a column is associated to each boosting method, a column to the instance tackled, and a last column to the performance

obtained. Based on this table, a set of (10 by default) regression trees is built. In each tree, the root is associated to all the configurations present in the table and each branch corresponds to the activation or not of a boosting method. In this way, each node is associated to a subset of the configurations of the input table that have in common the activation or the non activation of a subset of methods. The branching of a node is done provided that the corresponding subset of configurations has cardinality of at least 10 (SMAC default setting). Given a tree, the challenger performance is predicted as the average of the configurations associated to the only leaf that would have included the challenger if it had been one of the configurations in the input table.

Note that different branching are possible depending on which and on in which order the boosting methods are considered. Hence, SMAC builds several trees and obtains several predictions for a challenger performance. Then, it estimates the performance of a challenger as a random variable with a Gaussian distribution whose mean and variance are, respectively, the mean and the variance of the predictions of the different trees. SMAC finally computes the expected improvement brought by the challenger as the expected value of the maximum between 0 and the difference of the mean performance of the incumbent across the instances tackled and this random variable [12]. The expected improvement is large for a challenger with either good predicted performance or high predicted uncertainty. Hence, SMAC automatically tackles the tradeoff between focusing on known good subsets of boosting methods and gathering more information about subsets of boosting methods with unknown performance.

A more detailed explanation of SMAC can be found in [10].

V. EXPERIMENTAL ANALYSIS

In this section we assess the performance and the robustness of RECIFE-MILP. To this end, we consider three case-studies representing traffic in three control areas in France: the Pierrefitte–Gonesse junction (*Gonesse*), the Lille–Flandres station (*Lille*) and a line section around the Rouen–Rive–Droite station (*Rouen*). As typically imposed for the application of real-time traffic management tools, we set the computation time available for the optimization to three minutes [18]. The implementation is done using IBM ILOG CPLEX Concert Technology for C++ (IBM ILOG CPLEX version 12.6 [11]).

Despite the collaboration with SNCF experts, we could not identify realistic different priorities to be assigned to the delay of different trains, and hence w_t is set to 1 for all trains. The route formation and release times are 15 and 5 seconds for all block sections, respectively.

For each control area, we consider a one-day timetable, and we create 30 random scenarios: 20% of trains, randomly selected, suffer a random delay between 5 and 15 minutes at their entrance in the infrastructure considered. We generate ten rtRTMP instances from each of these 30 scenarios by considering all the trains entering the infrastructure within an hour horizon, starting from ten randomly drawn time instants between 5:00 A.M. and 9:00 P.M.. We use these 300 instances ($\in \Pi \setminus \tilde{\Pi}$) for testing RECIFE-MILP in a sample of cases

TABLE I
CONFIGURATIONS SELECTED BY SMAC

configurations	activated boosting methods
<i>smac_G</i>	-
<i>smac_L</i>	Topology, M-decrease, Shift, Distant-trains
<i>smac_R</i>	Topology, M-decrease, Shift, Distant-trains, Earliest-start, LX-relation

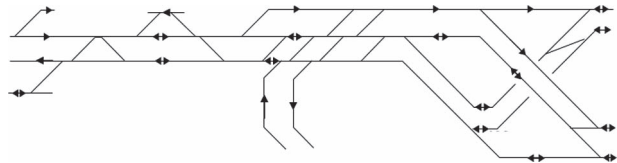


Fig. 1. Representation of the Gonesse control area.

which are representative of the whole day. For each control area, we create 50 further instances to be used in the algorithm configuration phase ($\bar{\text{I}}$) through the same procedure.

In the next subsections, we describe the algorithm configuration phase and the results obtained applying RECIFE-MILP to each control area with a time limit of three minutes. We refer to the configuration as *smac_G*, *smac_L* and *smac_R* for indicating that RECIFE-MILP is run with the boosting methods selected by SMAC on 50 Gonesse, Lille and Rouen instances, respectively. Moreover, we show the results obtained with no boosting methods (*plain*) and with all of them (*all*). The configuration *plain* is the one used in [16] which here we apply to the new MILP formulation described in Section II and we run for three minutes.

The benchmark we consider is the traffic management strategy currently in use in France (*cur*). This strategy gives priority to on time trains and performs no rerouting. In other words, whenever possible, this strategy allows a train which enters the infrastructure on time to carry on its whole trip without being delayed, even if this decision imposes to further delay the trains which do not enter the infrastructure as scheduled. Despite the sub-optimality of this traffic management strategy, it constitutes the main benchmark in the analysis since it allows the assessment of the improvement that optimization may bring in real world situations. According to the SNCF experts we collaborate with, such a benchmark is the most meaningful in the eyes of practitioners who actually deal with the current strategy every day. For a more academic comparison, we also analyze the solutions found by RECIFE-MILP with respect to the optimal solutions found by allowing longer runs to CPLEX when using the MILP formulation described in Section II (*optim*). Furthermore, we compare them with the optimal solution obtained when no rerouting is allowed (*fix*). Note that neither *optim* nor, to the best of our knowledge, any other algorithm can provide the proof of the optimality of rRTMP solutions in real-time.

A. Algorithm Configuration Phase

In this phase, we run SMAC independently for each control area. We impose the use of the Initialization method for all control areas, since a preliminary analysis showed that it is always helpful. We allow SMAC to run for five days for each control area. This computation time might appear long; however, it is acceptable since this procedure is to be run once for all before the beginning of the actual applications.

Table I reports the configuration selected for the three control areas. We observe that SMAC does not find any combination of boosting methods capable of significantly improving the performance of the plain RECIFE-MILP when tackling Gonesse.

Indeed, *smac_G* is equal to *plain*. This equivalence is most likely due to the fact that the performance of *plain* is already very good. Apparently, all the alternative configurations tested by SMAC were worse than *plain* at least for one instance. Differently, when dealing with the instances of Lille and Rouen, SMAC activates several methods: Topology, M-decrease, Shift and Distant-trains for *smac_L* and the same four methods plus the Earliest-start and the LX-relation for *smac_R*. Interestingly, the X-priority, the XY-relation and the Transitivity methods are never activated. Apparently, SMAC observed that they do not bring any advantage. The fact that X-priority is never selected indicates that CPLEX default branching strategy is already very good and any external intervention for modifying it does not bring substantial advantages. Similarly, the fact that XY-relation is never selected indicates that CPLEX default strategy for discarding equivalent sub-trees is also very good. Finally, the fact that Transitivity is never selected seems to indicate that the computational effort for dealing with the additional constraints introduced by this method is not compensated by an easy elimination of some branches in the search tree.

B. Gonesse Junction

Gonesse is a critical location with intense mixed traffic. Fig. 1 shows the control area considered. It includes 89 track-circuits, grouped into 174 block sections and 39 routes. The timetable of a week-day includes 340 trains crossing this control area: 120 high-speed and 129 conventional passenger trains, and 91 freight trains. The one-hour instances include between 9 and 29 trains (mean 14). Each train can use between 3 and 8 routes (mean 6), which translates into a MILP formulation (*plain*) with about 11 000 continuous variables, 9000 binary variables and 35000 constraints for an instance with 14 trains. We run the experiments on an Intel(R) Xeon(R) CPU E5-2665 0 @ 2.40 GHz, 16 cores, 32GB RAM. The mean optimal total delay propagation is 53 seconds.

Table II reports the results achieved by the different configurations of RECIFE-MILP: number of instances in which the optimal solution is found within the time limit; statistical significance of the difference w.r.t. *plain* according to the Wilcoxon rank-sum test with confidence level 0.95 (for each configuration, “b” if it is significantly better, “w” if it is significantly worse, “” if the difference between the two configurations is not statistically significant, i.e., if according to the experiments performed they are equivalent); mean percentage improvement w.r.t. *cur*; absolute value of the same difference; mean computation time. For ensuring the possibility of computing the percentage error also when *cur* finds a solution with null objective function value, we increase all

TABLE II
PERFORMANCE OF RECIFE-MILP WITH DIFFERENT CONFIGURATIONS
ON THE 300 INSTANCES AT GONESSE. COMPARISON WITH *cur*

configuration	# optima	stat	mean % imp.	mean imp. (sec)	mean time (sec)
<i>plain</i>	286		51%	175	17
<i>smac_G</i>	286		51%	175	17
<i>smac_L</i>	289		51%	175	14
<i>smac_R</i>	267	w	51%	174	13
<i>all</i>	255	w	49%	165	33

reference	# optima	stat	mean % imp.	mean imp. (sec)	mean time (sec)
<i>fix</i>	110	w	32%	132	0.
<i>optim</i>	300	b	51%	176	21

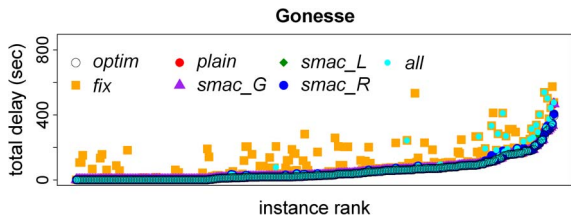


Fig. 2. Objective function value for the 300 instances at Gonesse.

values of 1 second. Hence, the percentage error is computed as: $\%error = (cur - \text{concerned configuration}) / (cur + 1)\%$. In all experiments, the $\%$ error is positive. A mean computation time lower than 180 seconds (the time limit), indicates that the solution is often proved to be optimal before this time.

The results show that RECIFE-MILP achieves extremely good performance disregard the configuration used, being at least 49% better than *cur*. The optimal solution is found by *plain* for 286 of the 300 instances, and the mean computation time is 17 seconds. Between *plain* and *optim*, the mean difference of improvement over *cur* is 1 second. As discussed in Section V-A, *smac_G* equals *plain*. In these experiments, *smac_L* achieves slightly better results, finding the optimal solution for three more instances, but the difference is not statistically significant. Instead, *all* is statistically worse than *plain*, but the number of optimally solved instances is still high.

Fig. 2 shows the results of each configuration and the optimal solutions with and without rerouting. The instances are ordered as a function of their optimal solution values. Apart from *fix*, all configurations get to rather indistinguishable results, also w.r.t. *optim*. Some minor differences are detectable only for the instances with the highest optimal total delay, but the results of all configurations are always quite close. The worst one is *all*, as remarked in Table II.

For Gonesse, then, RECIFE-MILP performs well, finding the optimal solutions in more than 95% of the cases. All the configurations chosen by SMAC achieve good results (in the worst case finding the optimum for 89% of the instances).

C. Lille–Flandres Station

Lille is a location where trains must necessarily cross for entering and exiting the station, which makes even the only scheduling decisions extremely critical. Just as a hint of this

criticality, the first-come-first-served approach, described in the framework of the Initialization boosting method, never finds a feasible solution on the instances considered: deadlocks always occur. In addition to this difficulty for the scheduling-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. 3. 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. Being a terminal station, rolling-stock is always used for both an arriving and a departing train. One of them may be a local movement connecting a depot to a platform. In total, 299 track-circuits compose 734 block sections and 2409 routes. The one-day timetable contains 509 trains: 41 high-speed and 416 conventional passenger trains, and 132 local movements (259 turnarounds, 8 joins and 10 splits). The one-hour instances include between 25 and 50 trains (mean 31). Each train can use between 1 and 72 routes (mean 11), which translates into a MILP formulation (*plain*) with about 26 000 continuous variables, 18000 binary variables and 76 000 constraints, for an instance with 28 trains. We run the experiments on an Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz, 24 cores, 32GB RAM. The mean optimal total delay propagation is 220 seconds.

Table III and Fig. 4 report the results on these instances in the same format described in Section V-B.

Here, the improvement w.r.t. *cur* is even higher than for Gonesse. In a sense, *fix* achieves poorer results than for Gonesse: the optimal solution corresponds to the fixed-route optimum in only 17 cases. However, this configuration allows a mean improvement of 55% over *cur*. The fixed-route optimum is always found very quickly, taking in the mean only 3 seconds. When rerouting is considered, as observed for Gonesse, it is not a good idea to apply simultaneously all boosting methods: the mean percentage improvement is 79%, compared to 80% of *smac_L* and *smac_R*; *all* is significantly worse than *plain*, finding the optimal solution in 195 cases compared to the 238 times of *plain*. Instead, here a configuration using a selection of boosting methods as *smac_L* and *smac_R* is significantly better than *plain*. The difference between *smac_L* and *smac_R* is not statistically significant in our experiments. Fig. 4 allows observing that the difference in the performance of the configurations is actually evident only in the upper part of the distribution of the instances considered in increasing order of optimal solution value.

In summary, also in Lille the performance of RECIFE-MILP is very good, with a detection rate of the optimal solution around 80% with the best configurations. Moreover, this performance is quite robust w.r.t. the configuration.

D. Line Around the Rouen–Rive–Droite Station

Rouen includes six stations, with two to six platforms, and one bifurcation. 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 location, together with the presence of mix traffic. The control

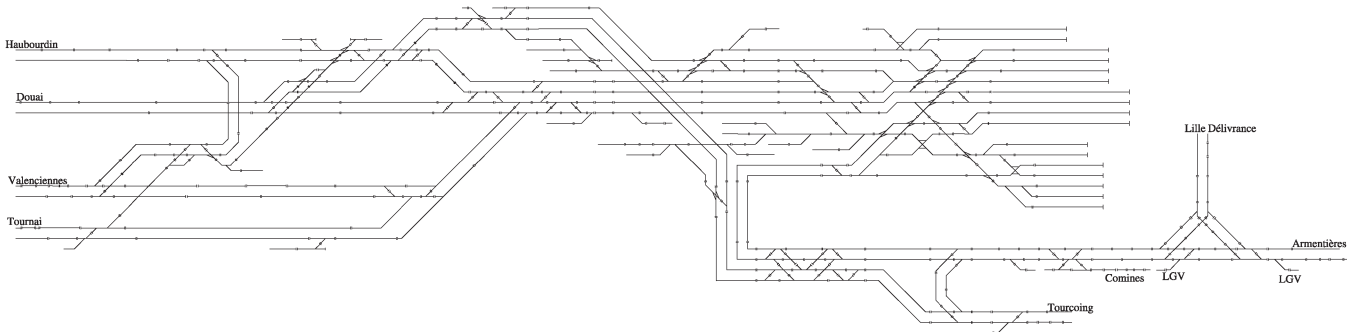


Fig. 3. Representation of the Lille control area.

TABLE III
PERFORMANCE OF RECIFE-MILP WITH DIFFERENT CONFIGURATIONS ON THE 300 INSTANCES IN LILLE. COMPARISON WITH *cur*

configuration	# optima	stat	mean % imp.	mean imp. (sec)	mean time (sec)
<i>plain</i>	238		79%	1511	101
<i>smac_G</i>	238		79%	1511	101
<i>smac_L</i>	254	b	80%	1525	78
<i>smac_R</i>	251	b	80%	1527	89
<i>all</i>	195	w	79%	1485	123

reference	# optima	stat	mean % imp.	mean imp. (sec)	mean time (sec)
<i>fix</i>	17	w	55%	1220	3
<i>optim</i>	300	b	82%	1642	17427

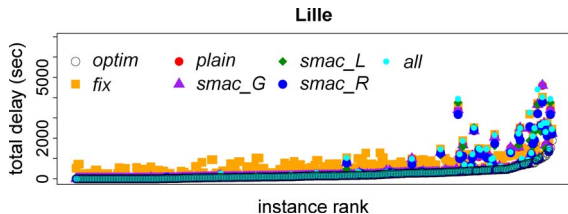


Fig. 4. Objective function value for the 300 instances in Lille.

area is depicted in Fig. 5. The 190 track-circuits compose 189 block sections and 11 347 routes. The one-day timetable includes 186 trains: 2 high-speed and 107 conventional passenger trains, 33 freight trains and 44 local movements. The one-hour instances include between 6 and 17 trains (mean 10). Each train can use between 1 and 192 routes (mean 45), which translates into a MILP formulation (*plain*) with about 117 000 continuous variables, 78 000 binary variables and 364 000 constraints, for an instance including 10 trains. We run the experiments on an Intel(R) Xeon(R) CPU X5650 @ 2.67 GHz, 12 cores, 24 GB RAM. The mean optimal total delay propagation is 60 seconds.

Table IV and Fig. 6 present the results achieved on the Rouen instances, as in Sections V-B and V-C. They show that the improvement w.r.t. *cur* is very high, even if slightly lower than for Lille. However, finding the optimal solution within the short computation time allowed is rather tough. The overall optimal solution (*optim*) never corresponds to the optimal solution without rerouting (*fix*). All configurations find the former in slightly more than a third of the instances. Their performance is similar, with a 70% improvement over *cur*.

The only significant difference w.r.t. *plain* is detectable for *smac_L*, which allows a further 6% improvement over *cur*. Hence, here, the results are quite robust w.r.t. the configuration of RECIFE-MILP: *smac_R* is statistically indistinguishable from *plain*; SMAC was not able to identify the best known configuration (*smac_L*) and it chose one that neither improves nor worsens the performance of the default one. Fig. 6 shows that the results get remarkably different from the optimum for about half of the instances, but no configuration is regularly better or worse than the others.

Intuitively, the source of complexity of the Rouen instances is the very large number of alternative routes per train. To support this intuition, Table V reports the results of the analysis run considering a subset of at most 25 routes for each train. The selection is random and there is no guarantee that the optimal solution is included in the search space. However, the performance of all configurations improves: the improvement w.r.t. *cur* is much higher than when considering all the possible routes and the optimal solution of the original instances is found much more often. Here, the configurations selected by SMAC for both Lille and Rouen are significantly better than *plain* due to the more effective exploration of the reduced search space. Indeed, with a search space of non-excessive size, the boosting methods applied are able to make the difference in terms of performance.

VI. CONCLUSION

In this paper we proposed RECIFE-MILP for the rtRTMP. RECIFE-MILP is a time-limited heuristic algorithm based on a MILP model that extends the one in [16]. It includes the consideration of further aspects of railway reality and it implements several performance boosting methods.

We assessed the performance of RECIFE-MILP in a thorough experimental analysis considering three French control areas with quite different characteristics. For each of them, we applied an automatic algorithm configuration procedure for selecting the most appropriate set of boosting methods to apply. With this analysis we could validate the three statements which we made in the introduction of this paper:

- 1) RECIFE-MILP is a very well performing algorithm for real-time applications: it often finds the overall optimal solution to instances, it improves remarkably over the

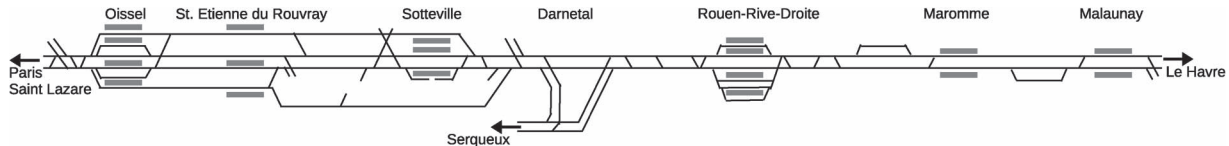


Fig. 5. Representation of the Rouen control area. The distances between stations are shrunk for ease of visualization.

TABLE IV
PERFORMANCE OF RECIFE-MILP WITH DIFFERENT CONFIGURATIONS ON THE 300 INSTANCES AT ROUEN. COMPARISON WITH *cur*

configuration	# optima	stat	mean % imp.	mean imp. (sec)	mean time (sec)
<i>plain</i>	123		70%	919	136
<i>smac_G</i>	123		70%	919	136
<i>smac_L</i>	137	b	76%	1028	133
<i>smac_R</i>	118		67%	912	141
<i>all</i>	114		68%	939	142

reference	# optima	stat	mean % imp.	mean imp. (sec)	mean time (sec)
<i>fix</i>	0	w	36%	827	1
<i>optim</i>	300	b	93%	1300	1020

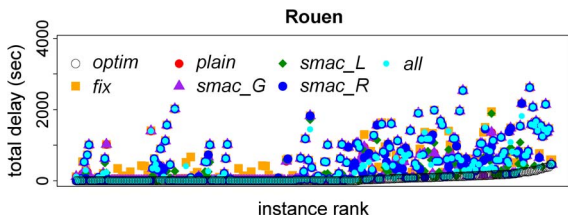


Fig. 6. Objective function value for the 300 Rouen instances.

TABLE V
PERFORMANCE OF RECIFE-MILP WITH DIFFERENT CONFIGURATIONS ON THE 300 INSTANCES AT ROUEN WITH A RANDOM SUBSET OF AT MOST 25 ROUTES FOR EACH TRAIN. COMPARISON WITH *cur*

configuration	# optima	stat	mean % imp.	mean imp. (sec)	mean time (sec)
<i>plain</i>	183		89%	1242	76
<i>smac_G</i>	183		89%	1242	76
<i>smac_L</i>	190	b	91%	1269	60
<i>smac_R</i>	192	b	91%	1272	61
<i>all</i>	184		90%	1251	69

optimal solution found with the routes indicated in the timetable, and it improves greatly with respect to the currently in use traffic management strategy;

- 2) it is robust to the characteristics of the infrastructure considered: it achieves high performance on all the three control areas, despite their different features;
- 3) it is robust with respect to the configuration used and it does not need to be specifically configured for each control area to be tackled. In particular, the best performance is obtained with the configuration including the boosting methods named Topology, M-decrease, Shift and Distant-trains in Section III.

Despite the good performance achieved, some difficulties still emerge when an extremely large number of routes are

available to each train, as in the case of the Rouen control area when some trains can use up to almost 200 alternative routes. Although the experts from the SNCF whom we interviewed declare that considering such a high number is not necessary and it will never be done in reality, the criteria to be used for the identification of the appropriate set of alternative routes have not been defined yet. Hence, in future work, we will focus on possible ways for appropriately reducing the number of alternative routes.

REFERENCES

- [1] R. Acuna-Agost, P. Michelon, D. Feillet, and S. Gueye, "A MIP-based local search method for the railway rescheduling problem," *Networks*, vol. 57, no. 1, pp. 69–86, Jan. 2011.
- [2] V. Cacchiani *et al.*, "An overview of recovery models and algorithms for real-time railway rescheduling," *Transp. Res. B, Methodol.*, vol. 63, pp. 15–37, May 2014.
- [3] 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," *Transp. Sci.*, vol. 45, no. 2, pp. 212–227, May 2011.
- [4] G. Caimi, M. Fuchsberger, M. Laumanns, and M. Luthi, "A model predictive control approach for discrete-time rescheduling in complex central railway station approach," *Comput. Oper. Res.*, vol. 39, no. 11, pp. 2578–2593, Nov. 2012.
- [5] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "A tabu search algorithm for rerouting trains during rail operations," *Transp. Res. B, Methodol.*, vol. 44, no. 1, pp. 175–192, Jan. 2010.
- [6] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "Optimal inter-area coordination of train rescheduling decisions," *Transp. Res. E, Logist. Transp. Rev.*, vol. 48, no. 1, pp. 71–88, Jan. 2012.
- [7] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "Dispatching and coordination in multi-area railway traffic management," *Comput. Oper. Res.*, vol. 44, pp. 146–160, Apr. 2014.
- [8] A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo, "Reordering and local rerouting strategies to manage train traffic in real-time," *Transp. Sci.*, vol. 42, no. 4, pp. 405–419, Nov. 2008.
- [9] F. Hutter, "Automated configuration of algorithms for solving hard computational problems," Ph.D. dissertation, Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada, 2009.
- [10] F. Hutter, L. Xu, H. H. Hoos, and K. Leyton-Brown, "Algorithm runtime prediction: Methods & evaluation," *Artif. Intell.*, vol. 206, pp. 79–111, Jan. 2014.
- [11] User's Manual for CPLEX, IBM Corporation, New York, NY, USA, 2012. [Online]. Available: <http://publib.boulder.ibm.com/infocenter/cosinfo/v12r2>
- [12] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimisation of expensive black box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, Dec. 1998.
- [13] R. M. Lusby, J. Larsen, M. Ehrgott, and D. M. Ryan, "A set packing inspired method for real-time junction train routing," *Comput. Oper. Res.*, vol. 40, no. 3, pp. 713–724, Mar. 2012.
- [14] O. Maron and A. W. Moore, "The racing algorithm: Model selection for lazy learners," *Artif. Intell. Rev.*, vol. 11, no. 1–5, pp. 193–225, Feb. 1997.
- [15] P. Pellegrini, G. Marliere, and J. Rodriguez, "Boosting the performance of a MILP formulation for railway traffic management in complex junctions," in *Proc. MT-ITS*, Dresden, Germany, 2013, pp. 419–428.
- [16] P. Pellegrini, G. Marliere, and J. Rodriguez, "Optimal train routing and scheduling for managing traffic perturbations in complex junctions," *Transp. Res. B, Methodol.*, vol. 59, pp. 58–80, Jan. 2014.

- [17] P. Pellegrini, G. Marliere, and J. Rodriguez, "Optimal train routing and scheduling in case of traffic perturbations: Improving solution time through parameter tuning," in *Proc. TRA*, 2014, pp. 1–9.
- [18] J. Rodriguez, "A constraint programming model for real-time train scheduling at junctions," *Transp. Res. B, Methodol.*, vol. 41, no. 2, pp. 231–245, Feb. 2007.
- [19] J. Rodriguez *et al.*, "RECIFE: Models and tools for analyzing rail capacity," (in French) *Recherche Transp. Sécurité*, vol. 95, pp. 19–36, 2007.
- [20] G. Theeg, U. Maschek, and O. Nasedkin, "Interlocking principles," in *Railway Signalling & Interlocking. International Compendium*, G. Theeg and S. Vlasenko, Eds. Hamburg, Germany: Eurail Press, 2009, pp. 61–112.
- [21] J. T. Krasemann, "Design of an effective algorithm for fast response to re-scheduling of railway traffic during disturbances," *Transp. Res. C, Emerging Technol.*, vol. 20, no. 1, pp. 62–78, Feb. 2012.
- [22] M. J. C. M. Vromans, R. Dekker, and L. G. Kroon, "Reliability and heterogeneity of railway services," *Eur. J. Oper. Res.*, vol. 172, no. 2, pp. 647–665, Jul. 2006.
- [23] J. Wales and M. Marinov, "Analysis of delays and delay mitigation on a metropolitan rail network using event based simulation," *Simul. Model. Pract. Theory*, vol. 52, pp. 52–77, Mar. 2015.



Paola Pellegrini received the Ph.D. degree in mathematics for economics from Ca' Foscari University, Venice, Italy, in 2007. She is currently a Researcher at the Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux (IFSTTAR), Université Lille Nord de France, Lille, France. She has been a Visiting Researcher at the University of Arizona (USA) and a Postdoctoral Researcher at the University of Trieste and Ca' Foscari University (Italy) and at the research institute IRIDIA (Belgium). She has participated in research

projects in the field of mathematical optimization, air traffic management, and railway transportation. She has published the results of her research studies in several scientific journal articles, book chapters, and conference papers.



Grégory Marlière received a degree in computer science from the University of Science and Technology of Lille (USTL), Lille, France, in 2001. He is an Engineer at the Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux (IFSTTAR), Université Lille Nord de France, Lille, France, in the Traffic Management Team. In his more than ten years of experience, he has participated in several research projects in the field of railway traffic management, capacity analysis, and safety. He has co-authored journal and

conference papers on these topics. He has developed a number of software tools for decision support and analysis in the framework of both the conventional railway traffic management system and the novel European Railway Management System.



Raffaele Pesenti received the Ph.D. degree in systems sciences at the University of Genova, Genova, Italy. He is a Full Professor of operations research at the Department of Management, Ca' Foscari University, Venice, Italy. His major area of research is management of complex systems with application in transportation and logistics. The interests in this field are devoted to the study of the strategic and analytical elements of the systems design and to the development of models and methods that may find application in the day-by-day operations. His works

appear on some of the highest impact journals in the field.



Joaquin Rodriguez received the Ph.D. degree in computer science at the University of Compiègne, Compiègne, France, in 1985. He is a Research Director at the Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux (IFSTTAR), Université Lille Nord de France, Lille, France. He is responsible for the traffic management team. He has 20 years of experience in coordinating industrial and academic research projects dealing with railway traffic management, capacity analysis, diagnosis, and safety. He has developed several decision support tools in transportation applications: real-time train circulation management in railway nodes, simulation of guideway transit systems, and model-based diagnosis of electronic devices.

developed several decision support tools in transportation applications: real-time train circulation management in railway nodes, simulation of guideway transit systems, and model-based diagnosis of electronic devices.