



HAL
open science

The potential of the routing selection problem in real-time railway traffic management

Marcella Sama, Paola Pellegrini, Andrea d'Ariano, Joaquin Rodriguez, Dario
Pacciarelli

► **To cite this version:**

Marcella Sama, Paola Pellegrini, Andrea d'Ariano, Joaquin Rodriguez, Dario Pacciarelli. The potential of the routing selection problem in real-time railway traffic management. 7th International Conference on Railway Operations Modelling and Analysis - Rail Lille, Apr 2017, Lille, France. 19p. hal-01524253

HAL Id: hal-01524253

<https://hal.science/hal-01524253v1>

Submitted on 17 May 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.

The potential of the routing selection problem in real-time railway traffic management

Marcella Samà^{a,1}, Paola Pellegrini^b, Andrea D'Ariano^a,
Joaquin Rodriguez^c, Dario Pacciarelli^a

^a Department of Engineering, Roma Tre University
Via della Vasca Navale 79, 00146 Roma, Italy
¹ E-mail: sama@ing.uniroma3.it

^b Univ Lille Nord de France, F-59000 Lille
IFSTTAR, COSYS, LEOST, F-59650 Villeneuve dAscq, France

^c Univ Lille Nord de France, F-59000 Lille
IFSTTAR, COSYS, ESTAS, F-59650 Villeneuve dAscq, France

Abstract

The real-time railway traffic management problem (rtRTMP) is the problem of solving time-overlapping conflicting track requests generated by disturbances during the normal course of daily operations. The size of an instance of the problem and the time required to solve it are particularly affected by the number of routing alternatives available to each train. The real-time train routing selection problem (rtTRSP) chooses a feasible routing subset for each train to use as input for the real-time railway traffic management problem. In particular, the validity of the problem and the impact of its solution have been analyzed in the literature. The analysis has been performed by using a specific mixed-integer linear programming model and the RECIFE-MILP solution approach for the rtRTMP. This paper generalizes such analysis by considering also a different model, objective function and solution approach for the rtRTMP. We consider the decision support system AGLIBRARY which solves an alternative graph model of the problem to minimize the maximum consecutive delay. The aim is to assess whether the selection of a subset of routings brings advantages for different solvers. We analyze how changes in the rtRTMP model have to be reflected in the rtTRSP and, if so, which modifications are required. The computational analysis is performed on two French infrastructures: the line around the city of Rouen and the Lille terminal station area. The analysis shows that solving the rtTRSP helps both solvers significantly, even if they are based on different models and algorithms.

Keywords

Train Re-routing and Re-scheduling, Train Routing Selection, Ant Colony Optimization, Alternative Graph, Mixed-Integer Linear Programming.

1 Introduction

In the last decades there has been a significant increase in railway traffic demand. This trend will most likely continue in the future and requires an adequate response by the infrastruc-

ture managers. Due to high costs or physical constraints, building new infrastructures is not always possible, thus requiring to use the already existing ones at their full capacity. For each day, a timetable is carefully planned in advance (Cacchiani and Toth, 2012). However, disturbances may happen in real-time, creating time-conflicting requests for the utilization of tracks by multiple trains. Traffic controllers are required to make train timing, ordering and routing decisions to appropriately solve these conflicting requests. Unfortunately, due to the real-time nature of the task and to the absence of ad-hoc decision support systems, traffic controllers can hardly estimate the overall effect of their decisions.

The real-time Railway Traffic Management Problem (rtRTMP) is the formalization of the problems faced by dispatchers: it consists in detecting and solving time-overlapping conflicting requests by multiple trains for the utilization of the same track(s), which make the timetable infeasible. In the literature, there is not a general agreement on how to model the rtRTMP, nor on the specific objective function to capture the quality of a solution. To model an infrastructure, two types of granularity are commonly used: *macroscopic*, where each resource corresponds to groups of block-sections (Törnquist and Persson, 2007), and *microscopic*, where each resource corresponds to a single block-section (Samà et al., 2017) or even to a single track-circuit (Caimi et al., 2012; Pellegrini et al., 2014), with possible combinations of the two for different parts of the infrastructure (Lamorgese and Mannino, 2015). To evaluate the quality of an rtRTMP solution, the objectives considered are often based on the services provided to the passengers (Corman et al., 2016; Sato et al., 2013; Schöbel, 2007), or on the minimization of the train delays propagation on the network. This second one, in particular, has been modelled as the minimization of the maximum consecutive delay (Samà et al., 2017), of the total (cumulative) delay (Meng and Zhou, 2011; Pellegrini et al., 2014) or as the minimization of the deviation from the timetable (Caimi et al., 2012).

Different advanced techniques to solve the rtRTMP can also be found in the literature (see, e.g., Fang et al. (2015); Takeuchi and Tomii (2005); Hansen and Pachl (2014)). Particularly, a stream of research successfully solves the scheduling problem to optimality by fixing the routing chosen in the timetable for all trains (Meng and Zhou, 2011; Törnquist Krasemann, 2012). Other approaches include routing decisions to better balance the use of infrastructure resources (Pellegrini et al., 2015; Samà et al., 2017). Samà et al. (2016) show how (to find a high quality solution to the rtRTMP) a partial routing flexibility is to be preferred to a complete routing flexibility: the alternative routings available for each train strongly affect the problem size and the required computation time. Most of the existing approaches limit the number of routing alternatives by using a given input and/or by not considering re-routing possibilities outside stations (Lamorgese and Mannino, 2015; Sels et al., 2014). Usually the considered routing subsets are defined by following traffic controllers suggestions and infrastructure managers directives, i.e., by considering alternative routings similar to the ones defined in the timetable. However, Samà et al. (2016) evidence how the specific routing subsets considered may have a great impact on the quality of the rtRTMP solution. This evidence motivates the need for a systematic study of the routing subset selection for each train and for the definition of the real-time Train Routing Selection Problem (rtTRSP).

The rtTRSP is the problem of choosing a routing subset for each train so that feasible combinations of train routings exist and good quality solutions for the rtRTMP can be rapidly found. In the rtTRSP, train timing and ordering decisions are neglected. The first optimization approach for the rtTRSP in the literature is the one in Samà et al. (2016).

The rtTRSP is modelled by using an N -partite graph and solved via an algorithm based on Ant Colony Optimization (ACO-rtTRSP) (Dorigo and Stützle, 2004). Specifically, ACO-rtTRSP is inspired by the ACO algorithm for the maximum clique problem (Solnon and Bridge, 2006a). The validity of this approach has been proven when the rtRTMP is microscopically modelled on a track-circuit level with a state-of-the-art mixed-integer linear programming (MILP) formulation (Pellegrini et al., 2014) and solved with RECIFE-MILP (Pellegrini et al., 2015), a decision support tool developed at IFSTTAR, considering as objective function the minimization of the total train delay.

This paper aims to provide a more general analysis on the impact that solving the rtTRSP has on the rtRTMP, regardless of the specific model considered for the rtRTMP, the objective function optimized and the rtRTMP solution approach used. In addition to the model by Pellegrini et al. (2014) solved with RECIFE-MILP, we consider the rtRTMP microscopically modelled on a block-section level by using the alternative graph formulation (Mascis and Pacciarelli, 2002), considering as objective function the maximum consecutive delay minimization. The problem is then solved with AGLIBRARY (D'Ariano et al., 2007; Corman et al., 2014; Samà et al., 2017), a decision support tool developed at Roma Tre University. We consider AGLIBRARY, the optimization core of ROMA (D'Ariano, 2009), and RECIFE-MILP, since these are the two rtRTMP solvers demonstrating the highest potential in the European project ON-TIME (Quaglietta et al., 2016). Using these two solvers implies also considering two different interlocking systems (Hansen and Pachl, 2014). In the route-lock route-release interlocking system used for AGLIBRARY, the utilization of a block-section locks all block-sections sharing a track-circuit with it, independently on the actual position of the train within the block-section itself. In the route-lock sectional-release interlocking system used for RECIFE-MILP, the utilization of a block-section locks the block-sections sharing with it a not-yet-released track-circuit.

In this paper, a thorough experimental analysis on two French railway infrastructures is proposed. Multiple instances are studied, where train entrance delays are considered. The purpose of the proposed analysis is three-fold:

- Demonstrate whether the selection of a subset of routings has a considerable advantage not only on RECIFE-MILP but also on AGLIBRARY;
- Evaluate whether a change in the rtRTMP objective function needs to be reflected on the rtTRSP one and/or on the estimation of component and link costs of the construction graph adopted to model the rtTRSP in Samà et al. (2016);
- Investigate possible adjustments to the ACO-rtTRSP algorithm regarding the parameters tuning and local search strategies when varying the objective function.

The rest of the paper is organized as follows: Section 2 describes the rtRTMP and gives a general idea on the assumptions made in the two models used in this paper; Section 3 shows the rtTRSP and how different rtRTMP objective functions are reflected on the rtTRSP; Section 4 presents the experimental analysis performed on the Rouen and Lille infrastructures; Section 5 outlines the paper conclusions and future research directions.

2 The real-time Railway Traffic Management Problem

Let us consider a railway network and the set of trains requiring to use this network in a certain time horizon. The input data of the rtRTMP consists of the expected entrance

time and the initial position in the network of each train traveling the network in the time horizon. Furthermore, the following additional information for these trains is also known: the planned passing, arrival and departure times at stations or at other relevant points, the available routing alternatives and the travel times required to traverse the network resources. The rtRTMP can be defined as the problem of detecting the time-overlapping conflicting requests done by multiple trains for the utilization of the same track resource(s) and solving them by using train timing, ordering and routing decisions so that a recovery plan is created where no deadlock exists and each resource is utilized by no more than one train at a time. We define the utilization of a resource as the sum of reservation, travel and clearing times of a train.

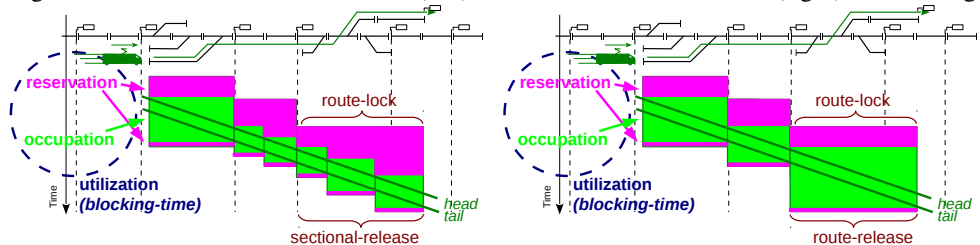
As previously stated, different models exist in the literature for the rtRTMP. Two in particular have been considered in this paper: the MILP formulation by Pellegrini et al. (2014) and the alternative graph model (Mascis and Pacciarelli, 2002) applied to railway systems by D'Ariano et al. (2007); Corman et al. (2014); Samà et al. (2017). In addition to other specific constraints in which the two models slightly differ, some common features considered are:

- *minimum headway times* to enforce a minimum separation between trains requiring to use the same network resources. These times are precisely computed depending on the microscopically represented interlocking system;
- *a fixed-speed model*, not computing the precise speed-variation dynamics for unplanned brakes and accelerations, which may be due to traffic.

While both the MILP formulation and the alternative graph model consider a microscopic granularity for the representation of the infrastructure and minimize the delay propagation, the two differ on the way they do so.

Considering the granularity of the representation of the infrastructure, in Pellegrini et al. (2014) each network resource is represented by a single track-circuit, while in Samà et al. (2017) by a single block-section. A *track-circuit* represents the most microscopic part of a network and a *block-section* is a sequence of track-circuits between two consecutive signals. This translates into considering two different *interlocking systems*, shown in Figure 1: route-lock sectional-release (in the left part of the figure) and route-lock route-release (in the right part of the figure).

Figure 1: Route-lock sectional-release (left) and route-lock route-release (right) interlocking



In the route-lock sectional-release interlocking system, when a train requires a particular block-section, it reserves all the track-circuits belonging to that block-section, preventing all other trains to use block sections sharing with it at least one track-circuit. When the train

has cleared a track-circuit, that track-circuit is released, together with all the block-sections including it and no other still-to-be-released track-circuits.

In the route-lock route-release interlocking system, again a train reserves all the track-circuits belonging to the block-section it needs to utilize, and this locks all the block sections sharing with the former at least a track-circuit. All block-sections are released when the train has cleared the block-section it utilizes, without any consideration on the actual position of the train within the block-section itself.

Considering the minimization of the delay propagation, the two models optimize two different objective functions. Pellegrini et al. (2014) minimizes the total train delays while Samà et al. (2017) the maximum consecutive delay.

The following subsections briefly present the MILP and alternative graph models. For a more in depth discussion on the two models we refer the interested reader respectively to Pellegrini et al. (2014) and Samà et al. (2017).

2.1 MILP formulation of Pellegrini et al. (2014)

When the rtRTMP is modeled as in Pellegrini et al. (2014), we simultaneously consider train timing, ordering and routing variables associated with the possible controllers decisions. Timing variables are non-negative continuous and account for the start and end times of track-circuit utilization and occupation by a train along a particular routing. Both train ordering and routing variables are binary, modeling respectively the order in which two trains utilize a common track-circuit and the choice of a single particular routing for a train, among its available alternatives.

The MILP formulation presents two distinct groups of constraints:

1. *Constraints on the traveling of the trains in the network.* Considering the available routings for each train and the necessity to select in the solution one and only one routing, these constraints impose the entrance times in the network, the arrival and departure times at stations or at other relevant points and the time required to traverse each track-circuit, according to each routing available;
2. *Constraints on the rolling stock configuration.* These constraints account for possible turnaround, join or split operations of trains. In particular, they model the minimum separation time required between the trains arrival and departure times, imposing that both need to take place on the same track-circuit to allow a specific rolling stock configuration;
3. *Constraints on the capacity of the network.* These constraints model the route-lock sectional-release interlocking system. Only one train at a time can utilize a block-section, unless the trains are involved in a rolling stock configuration change within the block-section itself.

We solve this MILP formulation by using the RECIFE-MILP algorithm developed by Pellegrini et al. (2015).

2.2 Alternative graph formulation of Samà et al. (2017)

When the rtRTMP is modeled as in Samà et al. (2017), it is considered as a job shop scheduling problem with flexible routes and additional practical constraints. We define an *operation*

o as the traversing of a network resource res (i.e., a block section) by a train a when using one of its available routings b . Let $G = (N, F, A)$ be the alternative graph composed of the following sets:

- $N = \{0, 1, \dots, *\}$ is the set of *nodes*, each associated to a particular operation o . Nodes 0 and * represent the start and the end operations of the schedule. To each node $o \in N$ we associate the start time h_o of operation o , corresponding to the entrance of the associated train in the associated resource, with $h_0 = 0$.
- F is the set of *fixed directed arcs* that model the sequence of operations to be executed by each train on each particular routing;
- A is the set of *alternative pairs*, each composed by two alternative directed arcs, that model the sequencing decision and safety separation constraints, considering the different available routing configurations.

The rtRTMP can be modeled as the disjunctive program:

$$\begin{aligned}
 & \min h_* \\
 & s.t. \\
 & h_e \geq h_o + w_{oe}^F + M(1 - y_{ab}) & \forall (o, e) \in F \\
 & h_k \geq h_o + w_{ok}^F + M(2 - y_{um} - y_{ab}) + M(1 - x_{(o,k),(e,d)}) & \forall ((o, k), (e, d)) \in A \\
 & h_d \geq h_e + w_{ed}^A + M(2 - y_{um} - y_{ab}) + M(1 - x_{(o,k),(e,d)}) & \forall ((o, k), (e, d)) \in A \\
 & \sum_{b=1}^{R_a} y_{ab} = 1 & a = 1, \dots, n
 \end{aligned} \tag{1}$$

where binary variable y_{ab} indicates if route a is chosen (1) or not (0) for train a , R_a is the number of routes for each train $a = 1, \dots, n$ and n is the number of trains traveling the network. When a route b is chosen for train a (i.e., $y_{ab} = 1$), each constraint related to route b and train a in F must be satisfied. Considering also the route m for the train u and the associated binary variable y_{um} , if $y_{um} = y_{ab} = 1$ and the routes of trains u and a use the same infrastructure resource res , a potential conflicting request exists on that resource, modeled by a pair in A , and a train ordering decision has to be taken.

We solve the rtRTMP when modeled with the alternative graph by using AGLIBRARY. In particular, we use the variable neighbourhood search metaheuristic developed by Samà et al. (2017) to iteratively select a routing for each train.

3 The real-time Train Routing Selection Problem

The rtTRSP is the problem of choosing for each train a subset of p routings among its available ones, when the alternatives for the train are larger than p . Being used as input for the rtRTMP, these subsets require to include feasible combinations of train routings and allow good quality solutions for the problem to be found quickly.

The rtTRSP is modeled by using the *construction graph* $G = (C, L)$ proposed by Samà et al. (2016). Figure 2 shows an example of construction graph. Each component $c_i \in C$ (a black bullet in Figure 2) represents an available alternative routing for a train. Each non-oriented link $l_{ij} \in L$ (a line between two black bullets in Figure 2) connects the two coherent components c_i and c_j it refers to. Two components are defined as *coherent* when they belong to different trains and the associated routings satisfy possibly existing rolling stock re-utilization constraints between the two trains. Such graph is n -partite, where n is

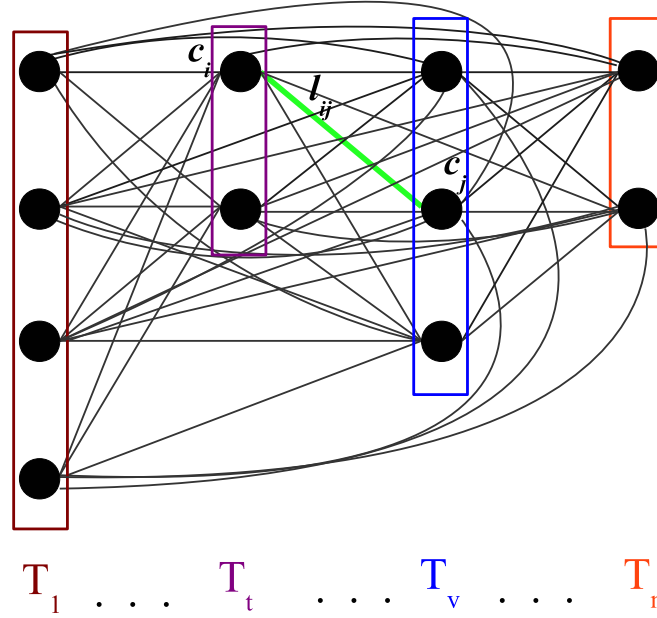


Figure 2: An example of construction graph $G = (C, L)$

the number of trains considered. Each partition (the rectangles in Figure 2), one for each train, represents all the available alternative routings of a given train. Finding one feasible train routing combination translates into finding a clique s of size n on the construction graph G , as shown in Formulation (2). The binary variable r_i indicates if component c_i has been selected ($r_i = 1$) or not ($r_i = 0$) in the train routing combination and the binary variable z_{ij} if link l_{ij} , and thus both c_i and c_j , has been selected ($z_{ij} = 1$) or not ($z_{ij} = 0$).

$$s = \left\{ \begin{array}{l} r \in \{0, 1\}^{|C|} \\ z \in \{0, 1\}^{|L|} \end{array} : \begin{array}{l} \sum_{c_i \in T_t} r_i = 1 \quad \forall T_t \subset C \\ \sum_{c_i \in C} z_{ij} = (n-1) r_j \quad \forall c_j \in C \\ z_{ij} = z_{ji} \quad \forall l_{ij} \in L \end{array} \right\} \quad (2)$$

The influence that selecting a particular train routing has on the rtRTMP solution is computed in terms of *potential delay*. The potential delay is an estimation of the delay propagation. When the potential delay is due to the routing in itself (i.e., if the alternative routing requires a longer travel time than the timetable one), it is represented by the cost u_i of component c_i . When the potential delay is due to the routing in combination with another one, it corresponds to evaluating the effect of the rise of conflicting requests between the two and it is represented by the cost w_{ij} of link l_{ij} . Let Γ be the set of all the n -cardinality cliques existing on G , for each $s \in \Gamma$ a cost f_s is associated. Finding a rtTRSP solution can be formulated as follows:

$$\begin{aligned}
 & \min \sum_{s \in \Gamma} f_s q_s \\
 & \text{s.t.} \\
 & \sum_{s \in \Gamma} q_s = m
 \end{aligned} \tag{3}$$

where the binary variable q_s , associated to a clique s on G indicates if a clique $s \in \Gamma$ is part of an rtTRSP solution S ($q_s = 1$) or not ($q_s = 0$). Let m be the number of feasible train routing combinations to be included in S , with $m \geq p$ necessary to cover multiple selections of the same component in different s .

The coupling between the rtTRSP and the rtRTMP solutions can be obtained through the components and links costs u_i and w_{ij} , representing the potential delay due to train routing and scheduling decisions respectively, and on how they are used to compute the clique cost f_s .

The cost u_i associated to component c_i represents the potential delay due to the routing decision. It is computed as the non-negative difference between the travel time of the corresponding train when it uses the assigned train routing and when it uses the timetable routing.

The cost w_{ij} represents the potential delay due to scheduling decisions between the pair of coherent train routing assignments c_i and c_j . We distinguish the cases in which the two trains travel in the same/opposite directions. If the two trains travel in opposite directions, we identify the train conflicting request (if any) with the highest number of common resources and we order the two trains in such a way that the consecutive delay is minimum. If the two trains travel in the same direction, we estimate the potential delay following a two-step procedure: (i) a set of common resources between the two train routing assignments is first selected and (ii) an estimation of the potential delay on the selected set of common resources is then computed. The set of common resources is built following one of these alternative strategies:

- **All** - the set contains all the resources for which train conflicting requests exist;
- **Min** - the set contains the resource k in each conflicting request with the *minimum maximum utilization time*;
- **Max** - the set contains the resource k in each conflicting request with the *maximum maximum utilization time*.

Using the set of resources computed in step (i), we estimate the potential delay in step (ii) following one of these three strategies:

- **Train** - We first define the potential delay for each train of the pair, and then define the potential delay of the pair as follows. For each selected resource, we compute the consecutive delay of each train when it waits for the other train. The potential delay for each train is the maximum among its consecutive delays over all the selected resources. The potential delay of the pair is the minimum between the potential delays of the two trains;
- **ResMin** - We first define the potential delay for each resource in the set of selected resources, we then define the potential delay of the pair as follows. For each selected

resource, we compute the consecutive delay of each train when it waits for the other train. The potential delay for each resource is the maximum between the consecutive delays of the two trains on that resource. The potential delay of the pair is the minimum among the potential delays for all selected resources;

- **ResMax** - The same as for ResMin, except that the potential delay for each resource is the minimum between the consecutive delays of the two trains on that resource, and the potential delay of the pair is the maximum among the potential delays for all selected resources.

After this procedure, an adjustment is performed in order to distinguish the case in which c_i and c_j do or do not have resources in common when the potential delay has been estimated as a non-positive quantity. In particular, if c_i and c_j have at least one common resource, some unpredicted train conflicting requests could still arise when solving the rtRTMP. We therefore set $w_{ij} = 1$. If the train routing assignments do not share resources, there will never be a conflicting request and $w_{ij} = 0$.

This cost computation can be applied to both models considered in this paper. In particular, when modeling the route-lock sectional-release interlocking system, each resource corresponds to a track-circuit, and the potential delay computations are made coherently. When modeling the route-lock route-release interlocking system, a resource is a block section, and the suitable utilization times are taken into account.

To complete the correspondence between the rtRTMP and the rtTRSP, we define the clique cost f_s based on the objective function solved in the rtRTMP. When the objective function optimized in the rtRTMP is the minimization of the total delay, we consider for a clique s its cost f_s^{tot} , computed as follows:

$$f_s^{tot} = \sum_{c_i \in C} u_i r_i + \sum_{l_{ij} \in L} w_{ij} z_{ij}. \quad (4)$$

In this case, the clique is evaluated based on all the components and all the links selected in the clique. In the rtRTMP, all the trains have an influence on the value of the solution obtained, in terms of both the routings the trains use and the orders in which they are scheduled on common resources. Thus in the rtTRSP we consider the influence that each component and each link has on the cost of the clique.

When the objective function minimized in the rtRTMP is the maximum consecutive delay, we consider for a clique s the cost f_s^{max} , computed as follows:

$$f_s^{max} = \max_{c_i \in C} \{u_i r_i + \max_{\delta(c_i)} \{w_{ij} z_{ij}\}\} \quad (5)$$

where $\delta(c_i)$ is the set of incident links l_{ij} on component c_i . When the maximum consecutive delay is minimized, a few trains may influence the value of the objective function. Thus, in the computation of the clique cost we try to identify which trains would be the most delayed ones. We make such assessment by identifying the trains for which the potential delay due to the routing assignment chosen and the worst potential delay due to possible scheduling decisions involving the routing assignment for the other trains is maximum. We thus search the routing assignments for which the sum of its components cost and its most costly incident link is maximum.

We solve the rtTRSP by using the ACO-rtTRSP developed in Samà et al. (2016). In the next section we present the computational results obtained. In Section 4.2 we study how to

compute the costs of components and links, based on the correspondence relation between the clique costs and the corresponding rtRTMP solution value.

4 Computational Results

This section presents different computational results on the rtTRSP and rtRTMP. In particular, Section 4.1 presents the instances used in the experiments, Section 4.2 the results concerning the selection on how the costs associated to components and links in the rtTRSP construction graph are computed, and Section 4.3 the results obtained when solving the rtRTMP considering as input routing subsets of different cardinalities returned by the rtTRSP solver.

The rtTRSP is solved by the ACO-rtTRSP algorithm of Samà et al. (2016). The rtRTMP is solved by the exact and heuristic approaches of RECIFE-MILP of Pellegrini et al. (2014, 2015) when modeled on a track-circuit level of detail and considering the total delay minimization. The rtRTMP is solved by using AGLIBRARY, in particular the variable neighbourhood search developed by Samà et al. (2017), when the rtRTMP is modeled on a block-section level of detail considering the maximum consecutive delay minimization.

All tests are performed with a Quad-Core Intel Xeon E5 3.7GHz processor with 32 GB RAM, under OS X 10.10.3.

4.1 Instances

The instances used in the computational experiments represent traffic in two French infrastructures. They are based on real-world data.

The first infrastructure is the 27-km railway line around the city of Rouen, shown in Figure 3. The line is composed of 190 track-circuits, 189 block sections, 11347 routings and several intermediate stations with up to six platforms. The Rouen line is traversed by 186 trains daily, each of them having up to 192 routings available.

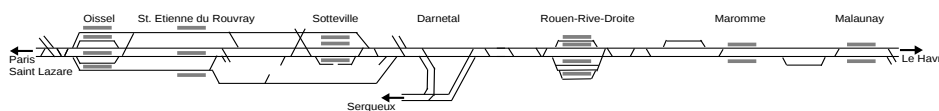


Figure 3: The Rouen network

The second infrastructure is the 12-km Lille station area shown in Figure 4. The station area is composed of 299 track-circuits, 734 block sections, and 2409 routings. The Lille station is a terminal station linked to national and international lines, with 17 platforms used by local, intercity and high speed trains. The Lille station area is traversed by 509 trains daily, each of them having up to 458 routings available.

We generated from a real timetable 5 scenarios for the Rouen case study and 2 scenarios for the Lille one as follows: In each scenario, 20% of the trains, randomly selected, are affected at their entry point by a random delay between 5 and 15 minutes. For each scenario, we generated 10 rtRTMP instances by considering all the trains that enter in the network within an hour, starting from 10 different time instants randomly taken during the day. In total, we obtain 50 rtRTMP instances for the Rouen case study and 20 instances for the Lille case study.

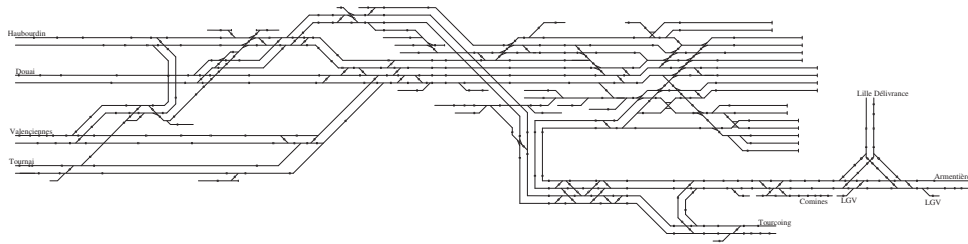


Figure 4: The Lille network

Looking at size of the Rouen instances, 11 trains require on average the use of the infrastructure, resulting in a construction graph G with an average number of components and links of $|C| = 491$ and $|L| = 99011$. Looking at size of the Lille instances, 29 trains require on average the use of the infrastructure, resulting in a construction graph G with an average number of components and links of $|C| = 4371$ and $|L| = 8777169$.

4.2 Choosing the rtTRSP costs

We proposed in Section 3 a two-step procedure to compute the rtTRSP cost w_{ij} representing the potential delay due to scheduling decisions between the pair of coherent train routing assignments c_i and c_j . For each step, we considered three alternative strategies. Here, we provide a computational analysis aiming to select the best rtTRSP cost computation among the combinations thus obtained.

We consider 30 Rouen instances. For each Rouen instance and for each cost computation, we generate 50 different rtTRSP solutions with routing subsets containing only one routing for each train, for a total of 1500 rtTRSP solutions for each cost computation. Starting from each of these 1500 rtTRSP solutions, we compute the optimal solution for the corresponding rtRTMP. Then, for each instance, we compute two rankings of the 50 solutions obtained: one is based on the rtTRSP solutions and on the related objective function; another is based on the rtRTMP solutions and on the related objective function. The two rankings are compared in order to find the cost computation that presents the minimum absolute difference with respect to the rtRTMP assessment. This difference is analyzed through the Wilcoxon rank-sum test with a confidence level of 0.95. We run this analysis for each of the two objective functions considered for the rtRTMP, making the relative adjustments to the rtTRSP objective. We observe that the two rankings are significantly different for all nine cost computations.

Figure 5 shows a boxplot comparison of the different rtTRSP cost computations available in terms of the absolute value of the difference between the rtTRSP and the rtRTMP rankings when considering the minimization of the total delay. Each boxplot represents 1500 values. We name each rtTRSP cost computation following the scheme $\langle set \rangle_ \langle estimation \rangle$, where the $\langle set \rangle$ refers to the criterion (All, Max, Min) used to select the resources to consider in the estimation (step i in Section 3) and $\langle estimation \rangle$ refers to the strategy (Train, ResMin, ResMax) used to estimate the potential delay on the selected common resources (step ii in Section 3). The thick horizontal line in the boxes represents the median of the distribution, the extremes of the boxes are the first and third quartiles, the dots are the outliers, and the whiskers show the smallest and the largest non-outliers in

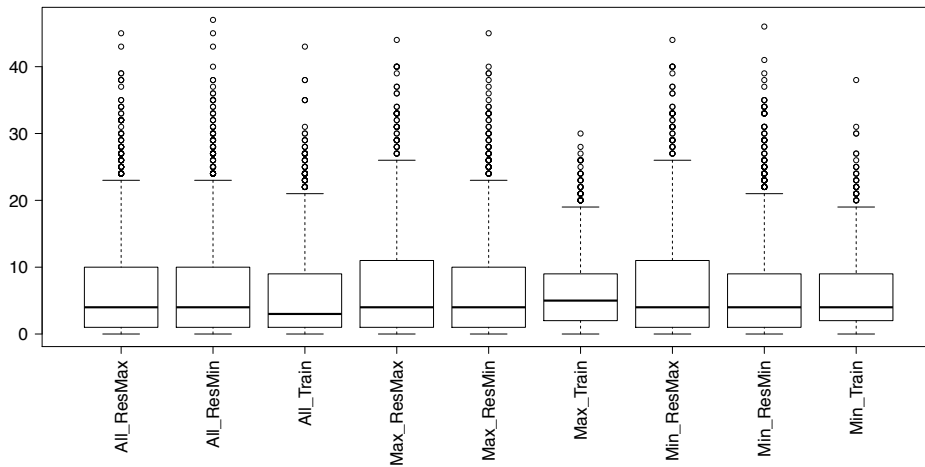


Figure 5: Boxplots of the difference in absolute value between the rtTRSP and rtRTMP rankings for the minimization of the total delay

the data-set. To complete the analysis, we consider the value of the *pseudo-median* of the distributions (Hollander and Wolfe, 1999) to obtain additional information on what would be the median of the distribution of the absolute difference of the two solution rankings.

The best minimum value of the pseudo-median is 6, found by cost computations *All_Train*, *Min_Train* and *Max_Train*. We note that a pseudo-median of 6 means that we can expect to obtain a ranking which differs of at most 6, in absolute terms, in 50% of the cases. Looking at Figure 5, the best cost computation appears to be *All_Train*. In fact, it has a median of 3 against 4 of *Min_Train* and 5 of *Max_Train*. Furthermore, it is the best until the 75th percentile of the distribution, where the three best cost computations become equivalent with an absolute difference of 9. In the next subsections, we consider *All_Train* set as the rtTRSP cost computation when minimizing the total train delays.

Figure 6 shows a boxplot comparison of the different rtTRSP cost computations available in terms of the of absolute value of the difference between the rtTRSP and the rtRTMP rankings when considering the minimization of the maximum consecutive delay. To select the cost computation, we follow the same procedure described for the first objective function. In this case, the best minimum value of the pseudo-median is 12, found by cost computations *All_Train*. Again, *Min_Train* and *Max_Train* have the second best value obtained, of 12.5. Looking at Figure 6, the best cost computation appears to be *All_Train*, since it has a median of 10 against 11 of both *Min_Train* and *Max_Train* and is the best until the 90th percentile of the distribution, where the three best cost computations become equivalent with an absolute difference of 28. In the next subsections, we consider *All_Train* set as the rtTRSP cost computation when minimizing the maximum consecutive delay.

Comparing the results obtained for the two objective functions, we see in both cases that the best cost computation is *All_Trains*. Looking at all resources possibly involved in conflicting requests and analyzing how the conflicting requests may affect each train appears to help in reaching a better assessment compared to the other options. However, looking at the differences between the two rankings, *All_Trains* better captures the total

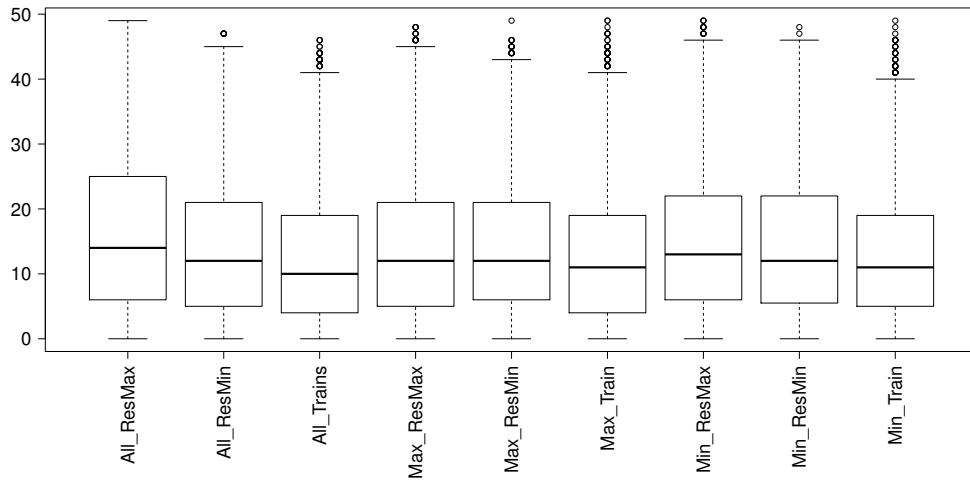


Figure 6: Boxplots of the difference in absolute value between the rtTRSP and rtRTMP rankings for the minimization of the maximum consecutive delay

delay objective function than the maximum consecutive delay. This may be due to the way in which the problem is actually modeled and solved. The construction graph allows for a more general view of the situation. The evaluation of the potential delays due to scheduling decisions, in particular, accounts only for the relation between two trains, not completely capturing how the delay propagates. However, counting how each pair of trains interacts helps counterbalancing the discrepancies between the two problems. Instead, for the minimization of the delay propagation, looking only at one specific potential conflicting request leaves more room for higher differences between the value of the solution found by the rtTRSP and the one of the rtRTMP.

4.3 Computational Experiments

This section presents the computational results obtained to analyze the benefit of solving the rtTRSP in terms of the performance obtained for the rtRTMP. The tests have been conducted on 20 instances for the Rouen infrastructure and 20 instances for the Lille one. In these experiments, we compare two different approaches used to solve the rtTRSP in order to generate the input for the solution of the rtRTMP:

- **ACO-rtTRSP:** An rtTRSP solution with p routing alternatives for each train is computed by the ACO-rtTRSP algorithm with a time limit of 30 seconds. This rtTRSP solution is given as an input for the solution of the rtRTMP. A computation time limit of 150 seconds is set for the rtRTMP solver. The best ACO-rtTRSP routing for each train is set as its default routing alternative;
- **ALL ROUTINGS:** All routing alternatives are given as input to the rtRTMP solver. A computation time limit of 180 seconds is set for the rtRTMP solver. The timetable routing is set as the default routing alternative for each train.

Table 1: Results obtained for the Rouen instances when solving the rtRTMP minimizing either the maximum consecutive delay (AGLIBRARY) or the total delay (RECIFE-MILP)

Approach	p	AGLIBRARY		RECIFE-MILP		
		Max Delay (sec)	Time Best (sec)	Total Delay (sec)	Time Best (sec)	Tot Time (sec)
ACO-rtRTMP	10	101.8	22	74.3	3	5
	20	107.0	8	74.3	8	12
	30	105.0	2	75.1	15	31
	40	100.2	3	73.5	17	29
	50	100.0	3	74.1	11	37
	60	100.0	2	73.5	19	41
	70	99.8	11	74.3	16	52
	80	99.8	8	74.3	13	35
	90	99.8	7	74.6	23	57
	100	99.7	5	77.1	19	50
	110	99.9	14	76.8	26	46
	120	99.8	11	74.1	24	57
	130	99.8	12	73.7	23	47
	140	99.8	8	74.1	29	57
	150	99.7	15	76.8	33	62
	160	99.7	12	74.1	19	57
	170	99.8	16	74.7	28	58
	180	99.8	13	80.1	32	58
	190	99.8	13	76.2	27	55
All Routings	192	100.6	31	76	42	70

Table 1 presents the results obtained for the Rouen instances. Column 1 specifies the solution approach used for the rtTRSP, Column 2 states the cardinality of the routing subsets passed to the rtRTMP, Columns 3-4 report the average value obtained by using AGLIBRARY when minimizing the maximum consecutive delay for the rtRTMP and the average computation time at which the best solution is found, Columns 5-7 show the average value obtained by using RECIFE-MILP when minimizing the total delay for the rtRTMP, the average time at which the best solution is found and the average total computation time. Indeed, RECIFE-MILP uses a truncated exact approach and, thus, may be able to certify the optimality of a solution. If it does so within the available computation time, it stops immediately. AGLIBRARY instead solves the problem via a meta-heuristic, thus always using all the computation time available.

The results show how, in terms of objective function value, for the Rouen instances solving the rtTRSP is more helpful when minimizing the total delay. For the minimization of the maximum consecutive delay, similar solutions are found when considering routing subsets with $p > 30$. However, combining ACO-rtTRSP with AGLIBRARY allows the computation of a good quality solution in a shorter computation time compared to the stand-alone AGLIBRARY (All Routings). We observe that it is not possible to directly compare the objective function of the two solvers, since they use different interlocking systems and optimization algorithms. In fact, the route-lock route-release implies a larger headway be-

tween trains than the route-lock sectional-release. This has an impact both on the solution assessment and on the search process (Pellegrini et al., 2014). Furthermore, we use an exact method in RECIFE-MILP, while we use a metaheuristic algorithm in AGLIBRARY. In future research, we will study more closely the specific behaviour of AGLIBRARY and RECIFE-MILP on these instances and others.

Table 2 presents the results obtained for the Lille instances. The results are presented with the same structure used for Table 1.

From the results of Table 2, the Lille instances appear more difficult than the Rouen ones. Overall, using ACO-rtTRSP is useful for both types of instances and for both solvers.

Table 2: Results obtained for the Lille instances when solving the rtRTMP minimizing either the maximum consecutive delay (AGLIBRARY) or the total delay (RECIFE-MILP)

Approach	p	AGLIBRARY		RECIFE-MILP		
		Max Delay (sec)	Time Best (sec)	Total Delay (sec)	Time Best (sec)	Tot Time (sec)
ACO-rtTRSP	10	342	39	238	74	82
	20	262	88	380	96	108
	30	229	91	532	107	130
	40	215	92	635	84	128
	50	217	91	638	81	142
	60	207	104	589	72	142
	70	207	103	654	81	142
	80	214	87	622	66	135
	90	182	95	663	69	137
	100	196	91	690	66	136
All Routings	458	265	30	717	76	173

For the minimization of the maximum consecutive delay, with a cardinality $p > 10$, ACO-rtTRSP reaches, on average, better solutions than All Routings. The bad performance obtained with $p = 10$ were to be expected, considering the difference in the solution rankings between the rtTRSP and the rtRTMP previously analyzed in Section 4.2. Routing subsets with higher cardinalities mitigate the imprecision of the costs estimation in perfectly capturing how the delay may propagate due to scheduling decisions. However, with a too high cardinality as in All Routings, finding the best combination and exploring the search space available becomes a very difficult task. This is noticeable looking at the value of the average time at which the best solution is found. The search space of All Routings is so large that its efficient exploration is hardly possible, and the algorithm is able to use only part of the computation time available to improve the solution. Eventually, the algorithm gets stuck in a local minimum and, even with the shaking and restart techniques used to differentiate the search, the large number of possibilities affects the performance.

For the total delay minimization, solving the rtRTMP considering as input only the routing subsets obtained by ACO-rtTRSP returns, on average, better solutions than solving the problem with all routings. In this case, the best cardinality for the routing subsets appears to be $p = 10$. In fact, increasing the cardinality makes the problem more difficult to solve. Not all the good quality solutions found with $p = 10$ are still found with $p = 20$, even if the best routing combination is often still present in the larger routings subsets. The

edge given to RECIFE-MILP by ACO-rtTRSP is not limited to the value of the solutions found. Also, smaller cardinalities speed up the solution process and the solver more often finds the optimal solution for the instances with the routing subsets. This is particularly evident comparing the Time Best and Total Time values. The difference between the two values grows at the growing of the subset cardinality. In particular, the Time Best remains on average approximately similar for different cardinalities, but the value of the actual best solution found is often very different, showing how the problems becomes more difficult to solve.

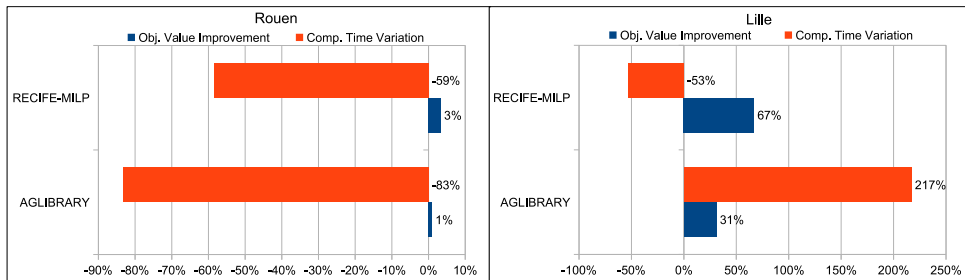


Figure 7: Barplots showing the average objective function value improvement and the average computation time variation for the Rouen (left) and Lille (right) instances

Figure 7 presents a barplot that summarized the results of Tables 1 and 2, highlighting the potential of combining the ACO-rtTRSP with AGLIBRARY or RECIFE-MILP. The best results obtained when the rtTRSP is solved are compared to the ones when all the available routings are considered. We consider the cardinalities $p = 40$ and $p = 10$ ($p = 100$ and $p = 90$) for RECIFE-MILP (AGLIBRARY), respectively, in case of the Rouen and Lille instances.

In Figure 7, the computation time variation is computed as follows: $(\text{computation time of All Routing } rtRTMP - \text{computation time of ACO-rtTRSP and } rtRTMP) / (\text{computation time of All Routing } rtRTMP)$, while the objective function improvement is computed as follows: $(\text{objective function value of All Routing } rtRTMP - \text{objective function value of ACO-rtTRSP and } rtRTMP) / (\text{objective function value of All Routing } rtRTMP)$.

In the experimental results on the Rouen network, the main advantage for both solvers is a considerably reduced computation time. There is also a minor advantage in terms of improvement of the solution quality. In the experimental results on the Lille network, the main advantage for both solvers is a significant improvement in the objective function value. For the RECIFE-MILP solver, there is also a considerable reduction in terms of the computation time. Differently, the improvement in solution quality is obtained, on average, by the AGLIBRARY solver in a larger computation time (but always within the maximum computation time given to the solver).

5 Conclusions and further research

Starting from the conclusion obtained in Samà et al. (2017), this paper investigates how using the ACO-rtTRSP helps in better solving the rtRTMP. An initial analysis has been performed by using the specific mixed-integer linear programming model and the RECIFE-MILP solution approach for the rtRTMP developed by Pellegrini et al. (2014, 2015). In this

paper such analysis is also performed for the alternative graph model and the AGLIBRARY solution approach for the rtRTMP (Samà et al., 2016).

Since different objective functions are optimized by the different solvers, we generalized the ACO-rtTRSP to deal with both objectives. From the computational results obtained in this paper, the clique model for the rtTRSP and the estimation of components and clique costs appear to be better in capturing the total delay objective function than the maximum consecutive delay one. For both solvers, limiting the number of routings available for each train, via the ACO-rtTRSP, helps the solution process of the rtRTMP, in terms of improving the solution quality and/or reducing the computation time. Comparing the impact of the ACO-rtTRSP for the two solvers, the improvement is higher for RECIFE-MILP.

Future research needs to look at improving the link between rtTRSP and rtRTMP. Although solving the rtTRSP with the proposed algorithm helps during the solution process of the rtRTMP, a better correspondence between the two problems is likely to be even more helpful. This better correspondence may be found through a better evaluation of the costs of the construction graph components and links. Also, a more extended analysis should be performed in order to pinpoint the best cardinality to be used for the routing subsets. Furthermore, we will focus on the specific behaviour of AGLIBRARY and RECIFE-MILP on these and other instances. Future research will be dedicated to study and identify how the difference in the objective functions, models and algorithms used by the two solvers affects their performance when the ACO-rtTRSP is used. Also, we are interested into analyzing the performance of each solver on its own and by comparing them.

Acknowledgements

We acknowledge the support received from Ing. Emanuele De Vico for his help in the generation of the instances.

References

- Cacchiani, V., Toth, P., 2012. Nominal and Robust Train Timetabling Problems. *European Journal of Operational Research* **219** (3) 727–737.
- Caimi, G., Fuchsberger, M., Laumanns, M., Lüthi, M., 2012. A model predictive control approach for discrete-time rescheduling in complex central railway station approach. *Computers and Operations Research*, 39 (11), 2578–2593.
- Corman, F., D’Ariano, A., Pacciarelli, D., Pranzo, M., 2014. Dispatching and coordination in multi-area railway traffic management, *Computers and Operations Research* **44** 146–160.
- Corman, F., D’Ariano, A., Marra, A.D., Pacciarelli, D., Samà, M., 2016. Integrating Train Scheduling and Delay Management in Real-time Railway Traffic Control, *Transportation Research Part E*, DOI: 10.1016/j.tre.2016.04.007
- D’Ariano, A., 2009. Innovative Decision Support System for Railway Traffic Control. *IEEE Intelligent Transportation Systems Magazine* **1** (4) 8–16.
- D’Ariano, A., Pacciarelli, D., Pranzo, M., 2007. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research* **183** (2) 643–657.
- Dorigo, M., Stützle, T., 2004. Ant Colony Optimization, *MIT Press, Massachusetts Institute of Technology, Cambridge*.

- Fang, W., Yang, S., Yao, X., 2015. A Survey on Problem Models and Solution Approaches to Rescheduling in Railway Networks. *IEEE Trans. on Intelligent Transportation Systems* **16** (6) 2997–3016.
- Hansen, I.A., Pachl, J., 2014. Railway timetabling & operations: analysis, modelling, optimisation, simulation, performance evaluation. *Eurailpress, Hamburg, Germany*.
- Hollander, M. and Wolfe, D. A., 1999. Nonparametric Statistical Methods, Second Edition, *New York: John Wiley & Sons*.
- Lamorgese, L., Mannino, C., 2015. An exact decomposition approach for the real-time Train Dispatching problem. *Operations Research*, **63** (1) 48–64.
- Mascis, A., Pacciarelli, D., 2002. Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research* **143** (3) 498–517.
- Meng, L., Zhou, X., 2011. Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. *Transportation Research Part B* **45** (7) 1080–1102.
- Pellegrini, P., Marlière, G., Rodriguez, J., 2014. Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Research Part B* **59** (1) 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. *IEEE Transactions on Intelligent Transportation Systems* **16** (5) 2609–2619.
- Quaglietta, E., Pellegrini, P., Goverde, R.M.P., 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. *Transportation Research Part C* **63** (1) 23–50.
- Sato, K., Tamura, K. Tomii, N., 2013. A MIP-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. *Journal of Rail Transport Planning Management*, 3 (3), 38–53.
- Samà, M., Pellegrini, P., D’Ariano, A., Rodriguez, J., Pacciarelli, D., 2016. Ant colony optimization for the real-time train routing selection problem. *Transportation Research Part B*, 85 (1), 89–108.
- Samà, M., D’Ariano, A., Corman, F., Pacciarelli, D. 2017. A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations. *Computers and Operations Research*, 78, 480–499
- Schöbel, A., 2007. Integer Programming Approaches For Solving The Delay Management Problem. In *Robust And Online Large-Scale Optimization*, LNCS, 4359, 145–170.
- Sels, P., Vansteenwegen, P., Dewilde, T., Cattrysse, D., Waquet, B., Joubert, A., 2014. The train platforming problem: The infrastructure management company perspective. *Transportation Research Part B* **61** (1) 55–72.
- Solnon, C., Bridge, D., 2006. An Ant Colony Optimization Meta-Heuristic for Subset Selection Problems. In *System Engineering using Particle Swarm Optimization*. Nova Science publisher, 7–29.
- Takeuchi, Y., Tomii, N. 2005. Robustness Indices for Train Rescheduling. In I.A. Hansen, F.M. Dekking, R.M.P. Goverde, B. Heidergott, & L.E. Meester (Eds.), Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis. Delft, the Netherlands.
- Törnquist, J., Persson, J., 2007. N-tracked railway traffic re-scheduling during disturbances.

Transportation Research Part B , 41 (3), 342–362.

Törnquist Krasemann, J., 2012. Design of an effective algorithm for fast response to re-scheduling of railway traffic during disturbances. *Transportation Research Part C* **20 (1)** 62–78.