



Ant colony optimization for the real-time train routing selection problem



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

^a Roma Tre University, Department of Engineering, Section of Computer Science and Automation, via della Vasca Navale, 79, 00146 Rome, Italy

^b University of Lille Nord de France, Ifsttar, COSYS, LEOST, rue Élisée Reclus 20, 59666 Villeneuve d'Ascq, Lille, France

^c University of Lille Nord de France, Ifsttar, COSYS, ESTAS, rue Élisée Reclus 20, 59666 Villeneuve d'Ascq, Lille, France

ARTICLE INFO

Article history:

Received 7 October 2015

Revised 11 January 2016

Accepted 12 January 2016

Available online 26 February 2016

Keywords:

Real-time railway traffic management

Train scheduling and routing

Meta-heuristics

ABSTRACT

This paper deals with the real-time problem of scheduling and routing trains in a railway network. In the related literature, this problem is usually solved starting from a subset of routing alternatives and computing the near-optimal solution of the simplified routing problem. We study how to select the best subset of routing alternatives for each train among all possible alternatives. The real-time train routing selection problem is formulated as an integer linear programming formulation and solved via an algorithm inspired by the ant colonies' behavior. The real-time railway traffic management problem takes as input the best subset of routing alternatives and is solved as a mixed-integer linear program. The proposed methodology is tested on two practical case studies of the French railway infrastructure: the Lille terminal station area and the Rouen line. The computational experiments are based on several practical disturbed scenarios. Our methodology allows the improvement of the state of the art in terms of the minimization of train consecutive delays. The improvement is around 22% for the Rouen instances and around 56% for the Lille instances.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

In the last decade, traffic demand in transports in general and railways in particular has significantly grown. To appropriately address this growth, railway undertakings and infrastructure managers face the challenge of expanding their offer. The difficulties in building new infrastructures due to high costs or physical obstacles translate into the need to utilize the already existing ones at their full capacity. To maintain a satisfactory quality of service and reduce passengers' inconvenience (Ginkel and Schöbel, 2007), it is necessary to manage precisely and efficiently the railway traffic in case of unexpected disturbances and disruptions that may affect the normal course of daily operations. Still, few decision support tools are available to help dispatchers in minimizing the impact of these events. Currently, dispatchers intervene manually in real-time, sometimes using pre-made contingency plans that, while helpful, cannot cover all the potential scenarios that may occur. Indeed, it is not easy to immediately judge the effects a particular decision may have. Solving a single event as best

* Corresponding author. Tel.: +39 0657333456.

E-mail address: sama@ing.uniroma3.it (M. Samà).

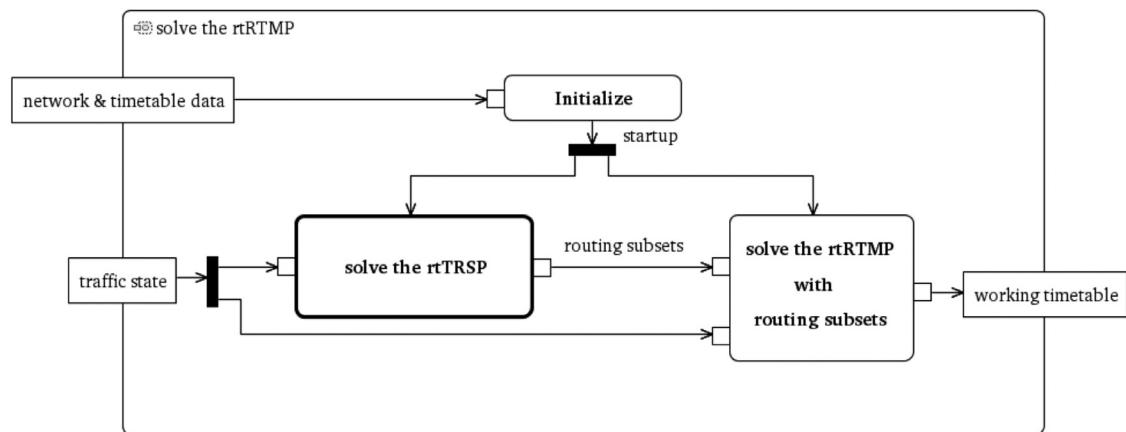


Fig. 1. SysML activity diagram representing how to solve the rtRTMP.

as possible could actually not be the most appropriate decision when considering traffic with a broader perspective. In fact, such a decision may indirectly generate further disturbances, propagating the delay in a snow ball effect.

The real-time Railway Traffic Management Problem (rtRTMP) is the problem of detecting and solving conflicting track requests by multiple trains during disturbed operations (Pellegrini et al., 2014). The rtRTMP is an NP-Hard problem where routing, ordering and timing decisions are made simultaneously. Moreover, the characteristics of the railway infrastructure and of the traffic flows, and in particular the number of routing alternatives for each train, strongly affects the time required to compute good quality solutions (Pellegrini et al., 2015).

Recent literature reviews show a rich stream of research focused on the rtRTMP (Cacchiani et al., 2014; Corman and Meng, 2015; Fang et al., 2015). Most of the existing approaches do not consider all set of possible routing alternatives present in the infrastructure for each train. A subset of routing alternatives is very often considered in the rtRTMP optimization. This subset is usually defined by taking into account the suggestions given by the dispatchers and by considering alternative routings similar to the one defined in the timetable. This is a reasonable assumption since the rtRTMP needs to be solved in real-time. However, the resulting rtRTMP optimization is a simplified routing problem and the computation of near-optimal solutions for the simplified problem does not necessarily correspond to near-optimal solutions for the original rtRTMP.

This paper studies the under-investigated problem of selecting the best subset of routing alternatives for each train among all possible alternatives. We call this problem the real-time Train Routing Selection Problem (rtTRSP). Fig. 1 shows a SysML activity diagram (SysML, 2015) in which the relation between the rtRTMP and rtTRSP solution activities is depicted. An activity diagram describes the sequence of actions to be performed to transform inputs into outputs. These inputs and outputs are represented as rectangles spanning the activity frame boundary. The actions are represented as rounded rectangles. An action begins as soon as all its required inputs are provided. In general, the rtTRSP is a subproblem of the rtRTMP: the rtTRSP solution allows the reduction of the size of the rtRTMP search space, which shall then be solved on the reduced space through one of the approaches existing in the literature. The input required to solve the rtRTMP are data on the infrastructure, the timetable and the traffic state at a reference time. The data on the network and the timetable are used to initialize both the “solve the rtTRSP” and the “solve the rtRTMP with the routing subsets” actions. After the initialization, the traffic state at a reference time is provided in order to perform the “solve the rtTRSP” activity by taking into account the current traffic information. This input is transmitted concurrently to the “solve the rtTRSP” and the “solve the rtRTMP with the routing subsets” actions. However, the latter can start only after the end of the former, since the required input “routing subsets” needs to be computed beforehand. The final output produced is a new working timetable, which includes all routing, ordering and timing information to be used in the operations.

The common practice of solving the rtTRSP is to follow some pre-determined directives given by the infrastructure managers on which routing subset should be considered for each train in real-time in order to help dispatchers when dealing with traffic disturbances. These directives are too general, while a case-to-case solution is required during operations. A dynamic selection of the routing subsets may intuitively improve the quality of the rtRTMP solution. However, the definition of the routing subsets requires to address the following trade-off. On the one hand, considering small subsets of all possible train routing decisions would reduce the number of variables to be tackled in the rtRTMP, and increasing the chances of finding good quality solutions in a short computation time. On the other hand, considering large subsets would increase the probability of preserving the optimal mix of train routings in the search space of the rtRTMP, but finding the global optimum may require a too high computation time.

This paper presents a general methodology to tackle the rtTRSP based on the scheme of Fig. 1. We propose an integer linear programming formulation of the rtTRSP with the objective of finding a good quality routing subset for each train, where the subset quality is defined as a function of the interactions among the routings selected for trains traveling in the infrastructure. Since a solution to the rtTRSP needs to be computed very quickly and the search space contains a very large

number of solutions, we solve the rtTRSP with an Ant Colony Optimization (ACO) meta-heuristic inspired by the foraging behavior of ant colonies (Dorigo and Stützle, 2004). The ACO meta-heuristic has already been applied to a number of complex problems in the railway domain, including rolling stock circulation (Tsuji et al., 2012), timetabling (Huang, 2006) and re-scheduling (Fan et al., 2012) problems.

In this work, the rtRTMP is modeled with the state-of-the-art formulation of Pellegrini et al. (2014) and solved with the exact and heuristic approaches proposed in Pellegrini et al. (2014, 2015). This is one of the most accurate formulations for the rtRTMP, in which the infrastructure is modeled at the level of track-circuit and of the interlocking system actually used in the practice. Several alternative routings are available for each train and the resulting rtTRSP is a challenging problem to be solved.

A thorough experimental analysis on two French case studies (the Lille terminal station area and the Rouen line) is proposed in order to assess the benefits of solving the rtTRSP to define the search space of the rtRTMP. Since practical-size rtRTMP instances present a too large number of variables, it is too time-consuming to consider all the routing alternatives for each train in the optimization model. For this reason, we analyze the performance of the heuristic approach *RECIFE-MILP* (Pellegrini et al., 2015) when using various routing subsets:

- all routing alternatives: this case corresponds to remove the rtTRSP from the solution process;
- a random selection of the routing subsets: we take a random solution of the rtTRSP;
- an optimized selection of the routing subsets: we take the best rtTRSP solution computed by ACO.

The computational results show that solving the rtTRSP is very useful, since reducing the number of routing subsets to be managed in the rtRTMP strongly improves the performance of the heuristic approach *RECIFE-MILP*. The improvement is larger when using the ACO meta-heuristic. All heuristic approaches are evaluated with a short computation time compatible with real-time application. We also compare the heuristic solutions with the global optimum computed by solving the formulation of Pellegrini et al. (2014) with a commercial solver, and evaluate the solutions computed by the heuristic approaches when varying the size of the routing subsets.

The paper is structured as follows: Section 2 presents a review of the rtRTMP literature. Section 3 defines the rtRTMP and the rtTRSP. Section 4 briefly reports the rtRTMP formulation (additional details are reported in Appendix) and introduces the rtTRSP formulation. Section 5 proposes methods to assign costs to the routing subsets in the rtTRSP. Section 6 describes the ACO meta-heuristic developed to solve the rtTRSP. Section 7 gives the results of the computational analysis on the two investigated test cases. Section 8 summarizes the conclusions of this work and outlines directions for further research on the rtTRSP and rtRTMP.

2. Literature review

Railway scheduling and routing problems deal with the efficient allocation of track capacity at strategic, tactical and operational levels (Lusby et al., 2011). On a strategic level, the problems deal with line planning and infrastructure construction and/or modification (Bussieck, 1998; Goossens et al., 2005; Scholl, 2005). On a tactical level, the problems deal with the creation of plans for the utilization of infrastructure and human resources, including timetable generation and platform planning in station areas (Cacchiani and Toth, 2012; Carey and Carville, 2003; Carey and Crawford, 2007; Sels et al., 2014; Zwaneveld et al., 2001). On an operational level, the problems deal with the recovery of the plans on the available infrastructure when disturbances and/or disruptions make them infeasible (Cacchiani et al., 2014; Corman and Meng, 2015; Fang et al., 2015). This section discusses recent state-of-the-art approaches on the rtTRSP and rtRTMP that are problems to be solved at the operational level.

The rtRTMP can be formulated in several ways: alternative graph model (Corman et al., 2009, 2010, 2011, 2014; D'Ariano et al., 2007, 2008), constraint programming (Rodríguez, 2007), mixed-integer programming (D'Ariano et al., 2014; Dessouky et al., 2006; Lamorgese and Mannino, 2015; Liu and Kozan, 2009; Meng and Zhou, 2011; Pellegrini et al., 2014; Törnquist and Persson, 2007), multicommodity flow models (Caimi et al., 2011; 2012), set-packing inspired models (Lusby et al., 2013), time-index models (Meng and Zhou, 2014). Further modeling approaches are discussed in the survey papers of Cacchiani et al. (2014), Corman and Meng (2015), Fang et al. (2015) and the recent book of Hansen and Pahl (2014). The various approaches differs in (i) the level of detail the authors formulate the rtRTMP and in (ii) the way their algorithmic methods manage the variables during the solution process.

Regarding (i), the main difference is the granularity used to model the infrastructure and traffic flows. Two main possibilities exist: *macroscopic* approaches model the resources as groups of block-sections (Dessouky et al., 2006; Lamorgese and Mannino, 2015; Mu and Dessouky, 2011; Törnquist and Persson, 2007), while *microscopic* approaches model each resource as a single block-section (Corman et al., 2010; D'Ariano et al., 2008; 2007) or a single track-circuit (Caimi et al., 2012; Pellegrini et al., 2014; Rodríguez, 2007). Considering track-circuits allows the modeling of two variants of interlocking systems: a route-lock route-release interlocking, in which 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; a route-lock sectional-release interlocking, in which the utilization of a block-section locks the block-section sharing with it a not-yet-released track-circuit. This second interlocking system can be considered only for microscopic models where a resource represents a single track-circuit. This paper deals with two French infrastructures, where the route-lock sectional-release interlocking is used

in practice. For this reason, we model the rtTRSP with the approach of Pellegrini et al. (2014), i.e., through a microscopic modeling of the track-circuits and the route-lock sectional-release interlocking system.

Regarding (ii), the key variables of the rtRTMP concern the routing alternatives for each train, its arrival and departure time at each station, the train ordering decisions on shared infrastructure resources. In general, the number of variables may be huge, so the computation time and memory usage to solve the rtRTMP may also be very high. To limit the complexity of the problem, different solution approaches have been considered. In Törnquist and Persson (2007), the ordering decisions concerning trains which are far away from each other, in terms of expected crossing time of common track sections, are considered fixed. In Caimi et al. (2011) and Rodriguez (2007), the rtRTMP is solved in two steps: first the routing to assign to each train is investigated and then the ordering decisions are taken. In Corman et al. (2011, 2014), D'Ariano et al. (2007, 2014), Liu and Kozan (2009), Meng and Zhou (2011) and Törnquist Krasemann (2012), the routing alternatives are not taken into consideration, setting to each train its timetable routing, and thus limiting the problem to timing and ordering decisions. In other approaches instead, the timetable routings are used to compute a first improvable solution. In Corman et al. (2010) and D'Ariano et al. (2008), this solution is bettered through an iterative procedure that, at each step, changes the routing for a certain number of trains and then finds new timings and orderings for the trains affected. The alternative routings are given as input to the iterative procedure (i.e., the rtTRSP is solved in a preliminary step). In Pellegrini et al. (2014, 2015), the objective function value of the initial solution is used as an upper bound to the problem. Other approaches instead look at different means to simplify their search for a solution. In Dessouky et al. (2006), deadlock avoidance checks are adopted to reduce the search space. In Caimi et al. (2012), the timing and routing decisions are combined through the definition of blocking-stairways, each one combining a routing and a speed-profile, selecting then few among a finite number of alternatives for each train. In Lamorgese and Mannino (2015) instead, the problem is tackled with a decomposition algorithm where a microscopic model is used in stations while a macroscopic one for the overall network. In Mu and Dessouky (2011), a simultaneous freight train routing and scheduling problem is formulated as a mixed integer programming model based on binary variables to explicitly model train ordering decisions and solved via some heuristic procedures based on clustering trains according to their entrance time in the network. A major drawback of this approach is the simplification of the topological structure of the network, since tracks are modeled macroscopically and can only allow trains traveling in a predefined traveling direction. In the rtRTMP, we need to deal with a microscopic representation of the network that allows all possible train routing alternatives. We also use a mathematical model based on binary variables for the train ordering and routing decisions. However, our main focus is not the modeling of the rtRTMP but the study of which train routing alternatives are more promising in order to efficiently solve the rtRTMP. In Meng and Zhou (2014), an alternative mathematical model is proposed in which the usage of resources is managed via cumulative flow variables instead of binary variables. This modeling approach has the advantage to limit the number of train ordering and routing variables. Another advantage compared to Mu and Dessouky (2011) is the capability to deal with any kind of network structure. The model of Meng and Zhou (2014) is solved via heuristics based on a problem decomposition in single train optimization sub-problems. This is a very efficient approach since each sub-problem has a relatively small size and time-dependent shortest path algorithms can be utilized to find good quality lower bounds. We believe that the identification of promising routing alternatives could be useful to further improve the performance of the proposed heuristics. However, a main drawback of the approach proposed by Meng and Zhou (2014) is the granularity of time in the timespace discretization of track occupancy. This approach applied to the rtRTMP studied in our paper could result in a huge variable space, since a level of detail of seconds is mandatory to model the occupation of each track-circuit.

Despite the effort done in decomposing the problem and developing strategies to efficiently manage the problem variables, the rtRTMP is still difficult to solve in a short computation time. Most of the existing approaches are able to provide good quality solutions only for simplified infrastructure configurations, a small number of trains, and simple traffic patterns. Therefore, there is a still need for developing more sophisticated and efficient methodologies to improve the solution process. This paper addresses one of the most under-investigated problems in the related literature that is the definition of a subset of routing alternatives for each train in the rtRTMP (i.e., the rtTRSP). In most of the approaches previously described, this is the starting point of the optimization process. However, the rtTRSP is often solved in a very simplified way, e.g., a number of alternatives is selected based on a-priori (Caimi et al., 2011) or random decisions (Pellegrini et al., 2015).

3. Problem description

A rail infrastructure is composed of sequences of *track-circuits*, which are grouped into *block-sections*. A track-circuit is the single part of an infrastructure where the presence of a train is automatically detected. A block-section is a sequence of track-circuits between two consecutive signals. These signals are used to communicate to train drivers the speed profile to adopt based on the status of the block-sections ahead. To provide clear signal to the driver, all the involved track-circuits must be reserved for the train itself before it can enter a sequence of block-sections, also allowing some additional time for preparing the routing, i.e., *routing formation*. After a train exits a track-circuit, its reservation is still active for the so-called *release time*, i.e., the time in which a track-circuit is released.

The travel of a train through a track-circuit or a block-section, according to the granularity of the model considered, is called *operation*. Depending on the infrastructure and the rolling stock composition, an operation requires a *travel time* to be completed, i.e., the time to process the infrastructure resource without encountering any other traffic. The clearing time is the time necessary for the tail of the train to clear the resource after the head has exited it. We name *utilization time* the

sum of reservation, travel time and clearing time by a train on an infrastructure resource. If a train starts its routing at null speed, its travel time on 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.

We call *network* the part of the infrastructure under study. A number of distinct alternatives of *routing* may exist in a network. A routing is a sequence of subsequent block-sections that leads from an entry point (i.e., the first resource of the routing) to an exit point (i.e., the last resource of the routing) in the network. A routing may include intermediate scheduled stops at stations. The travel times for the routing with intermediate stops includes the appropriate deceleration and acceleration times.

In a timetable, a particular routing is assigned to each train and it is called *default* or *timetable routing*. For each train, the scheduled departure and arrival times are defined for all the stations it stops at. A train is not allowed to depart from a station before its scheduled departure time, and it is late if arriving later than its scheduled arrival time. If the origin of the train is out of the network, its entrance time in the network can be computed based on the scheduled speed profile, and the train cannot enter the network earlier than this time. Similarly, if the destination of the train is out of the network, its exit time is computed and it is late if arriving at the limit of the network after it. Buffer times are carefully included in the timetable in order to absorb small perturbations. Even so, the timetable may become infeasible due to the arising of unexpected events during operation, which may cause disturbances, e.g., delays on the entrance times of trains in the network, and disruptions, e.g., the unavailability of one or more track sections of the infrastructure. Due to these events, two or more trains might generate a *conflicting request* for the same track-circuit or the same sequence of tracks-circuits, i.e., they may ask the same resource(s) at the same time. Each conflicting request causes unscheduled waiting times and thus longer travel times. To minimize the impact of disturbances and disruptions, an efficient working timetable must be quickly computed and implemented in real-time.

We call *primary delays* the delays directly caused by the unexpected events arising during rail operations. The propagation of the primary delays may generate additional delays, called *secondary delays*, due to conflicting requests of shared resources. Part of the latter delays does not depend on the decisions made during the computation of the working timetable for the following reasons. Given a network, a train entering with a delay may be unable to recover all or part of it before its scheduled arrival times or its exit. This is the case when a train is late even if it travels at its maximum speed and it is ordered before all the other trains on common track sections. The delay suffered by the train in this case is named *unavoidable delay*. We name *consecutive delay* of a train the difference between its secondary delay and its unavoidable delay. The consecutive delay is caused by the solution of conflicting requests, and is thus connected to the routing, ordering, timing decisions taken in the working timetable.

3.1. The real-time railway traffic management problem

We call *rtRTMP* (Pellegrini et al., 2014) the detection and solution of train conflicting requests during operations. Train routing, timing and ordering decisions are the problem variables to be fixed in order to achieve a feasible working timetable. A working timetable is feasible when a precedence between trains is chosen for each train conflicting request such that each resource is utilized by at most a train at a time and no deadlocks emerge in the network, i.e., there are no trains circularly waiting for each other in the network.

The *rtRTMP* considers a railway network and a set of trains required to traverse it in a given time period. For each train a single routing must be selected among a set of possible alternatives. Alternative routings must have the same entry and exit points unless they correspond to a station platform. In this case the other platforms of the same station may be alternative entry/exit points. Moreover, it is generally assumed that all alternative routings have to pass in the stations where the train is required to stop, according to its timetable. Each station operation has to be scheduled such that the train departure time is respected. The scheduling decisions are the train timing and ordering decisions.

Regarding the objective of the *rtRTMP*, train delays are usually minimized. However, this can be translated into different objective functions. While not the only ones existing, the objective functions related to train delays can be classified based on i) the type of delays considered, e.g., secondary (Meng and Zhou, 2014) or consecutive (Corman et al., 2010; D'Ariano et al., 2007); ii) what function of the delay is minimized, e.g., maximum (Corman et al., 2014; D'Ariano et al., 2014) or total (Pellegrini et al., 2015; Rodriguez, 2007); iii) the consideration of weights for the different types of trains, (Lamorgese and Mannino, 2015; Törnquist Krasemann, 2012). The *rtRTMP* studied in this paper minimizes the total consecutive delays with no distinction between train delays. These delays are influenced by the train scheduling and routing decisions, i.e., by the routing assigned to each train and the ordering and timing decisions on each resource.

3.2. The real-time train routing selection problem

We now introduce some preliminary definitions. A *train routing assignment* is the assignment of a routing to a train among the existing routing alternatives. The train routing assignments for a pair of trains t and v are *coherent* in two cases: i) there are no rolling stock constraints between the two trains; ii) the two trains have a rolling stock constraint and the train routing assignment of the first train has the exit point in the same resource in which the train routing assignment of the second train has the entry point. The latter case models, e.g., turnaround, join or split constraints. A combination

of train routing assignments is *feasible* if the train routing assignments for each pair is coherent. We call Γ the set of all feasible combinations of train routing assignments.

The real-time Train Routing Selection Problem (rtTRSP) is the problem of defining for each train a subset of routings selected among the existing routing alternatives available for each train. Each train routing assignment in the subset must be part of a feasible combination of train routing assignments. An rtTRSP solution S is therefore contained in Γ .

We let p be the maximum cardinality of the subset of routings selected for each train, with p predefined parameter. In this paper, the best value of p is investigated for each network under study. To assess the influence of the selection of p routing alternatives for each train on the original rtRTMP, the rtTRSP evaluates the subsets of routing alternatives by using the concept of *potential delays*: they are the estimation of the propagation of train delays in the network for the given train routing assignments. These estimations consider implicitly the train timing and ordering decisions. Finding the optimal solution S^* of the rtTRSP consists of selecting the train routing subsets with minimal potential delays.

The potential delays shall be defined in such a way that they take into account as much as possible the objective function of the rtRTMP, since the rtTRSP solution is an input data for the rtRTMP (as illustrated in the activity diagram of Fig. 1). Therefore, the objective function considered in the rtRTMP shall guide the choice of the objective function for the rtTRSP. In this work, potential delays are based on consecutive delays and the rtTRSP objective is the minimization of their sum.

4. Problem formulation

This section is divided into two parts. The first part presents a brief description of the mathematical formulation of the rtRTMP; a more detailed description is provided in Appendix. The second part introduces the mathematical formulation of the rtTRSP and provides an illustrative example.

4.1. The rtRTMP formulation

We consider the rtRTMP modeled using the MILP formulation presented in Pellegrini et al. (2014, 2015). The MILP formulation models the network microscopically where each resource represents a single track-circuit. A route-lock sectional-release interlocking system is employed. Timing decisions are modeled using non-negative continuous variables. In particular, they account for the start and end times of track-circuit utilization by the trains and the minimum and actual travel times of a track-circuit for a train along a particular routing. Binary variables model train routing decisions, as well as train scheduling decisions. For the routing decisions, the binary variables model the choice of a particular routing for a train, among its alternatives. For the scheduling decisions, the binary variables model the order in which two trains utilize a track-circuit belonging to both their routings.

The MILP formulation presents a series of constraints which can be classified in three distinct groups:

1. Constraints concerning the traveling of the trains in the network. Supposed that a train uses exactly one of its routing alternatives to traverse the network, these constraints impose the minimum entrance times in the network and the minimum departure times from stations, considering the scheduled times and possible primary delays. They also impose the minimum travel times on all track-circuits;
2. Constraints concerning the change of rolling stock configuration. In this model, turnaround, join or split of trains are taken into consideration. This translates into sets of constraints that impose time and space coherence between trains concerned by these changes of rolling stock configuration. In particular these constraints model the minimum separation time required between the trains arrival and departure. Both have to happen on the same track-circuit;
3. Constraints concerning the capacity of the network. These constraints model the route-lock sectional-release interlocking system. We remind that 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.

All three groups of constraints depend on the routing binary variables. In the first group, a set of constraints models how each train uses exactly one routing to travel in the network. Furthermore, the travel time on a track-circuit depends not only on the track-circuit itself, but also on the routing along which it is used. For example, a train with a particular routing could be required to brake on a track-circuit due to a switching requirement, while this may not happen for all the other routings of this train, even if they include the same track-circuit. Thus, the binary variables indicating the routing decision appear in the constraints concerning the travel times, setting them to 0 along all the not chosen routings.

In the second group of constraints, the relation between the routings of the arrival and the departure trains has to be satisfied. In particular, the last track-circuit of the arriving train routing has to be the same of the first track-circuit of the departing train routing.

In the third group of constraints, the use of the binary variables stating the routing decision is due to the fact that, indeed, the utilization time is a function of the travel time, which necessitates of these variables.

Since the routing binary variables appear in all groups of constraints, it becomes evident how diminishing the number of routing decisions strongly affects the size of the overall problem: not only the number of routing variables decreases, but also the variables of all timing decisions pertaining the discarded routings can be erased, together with the constraints on all three groups in which only those variables would have appeared.

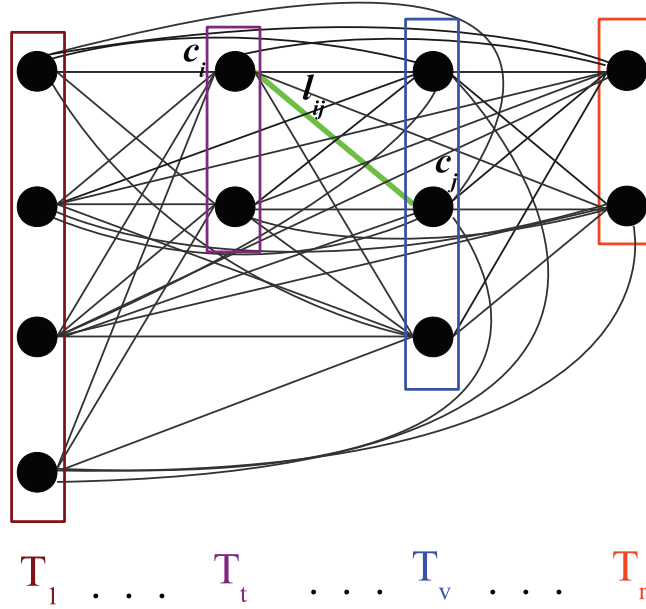


Fig. 2. Example of a construction graph $G = (C, L)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

The objective function considered in this paper for the rtTRMP is the minimization of the total consecutive delays suffered by trains at the exit points.

4.2. The rtTRSP formulation

We model the rtTRSP by means of a *construction graph* $G = (C, L)$, in which C is the set of components and L is the set of links. In Fig. 2, each component is depicted as a black circle, while each link is depicted as a non-oriented line between two black circles.

The set of components C is divided into n disjoint subsets of components. Each subset of components $T_t \subset C$ represents the routing alternatives for a train t and there are n trains in the network. A component $c_i \in T_t$ is a train routing assignment for train t , where $i = 1, \dots, |T_t|$.

Let us consider two components $c_i \in T_t$ and $c_j \in T_v$, with $t, v \in \{1, \dots, n\}$ and $t \neq v$, a link $\{c_i, c_j\} = l_{ij} \in L$ (highlighted in green in Fig. 2) exists if and only if the corresponding train routing assignments are coherent.

A clique s of size n in G represents a feasible combination of train routing assignments. The set of all possible cliques of size n in G models the set Γ of all feasible combinations of train routing assignments. A subset $S \subset \Gamma$ is an rtTRSP solution.

A clique $s \in \Gamma$ is defined as follows:

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

where r_i is a binary variable associated with component $c_i \in C$ and stating whether c_i has been selected in a feasible combination of train routing assignments ($r_i = 1$) or not ($r_i = 0$); z_{ij} is a binary variable associated with each link $l_{ij} \in L$ and stating whether link l_{ij} are selected ($z_{ij} = 1$) or not ($z_{ij} = 0$).

Constraints (1.a) ensure the selection of exactly one train routing assignment for each train t among its $|T_t|$ routing alternatives. Constraints (1.b) ensure the selection of exactly $n-1$ links incident to component c_j (i.e., $\sum_{c_i \in C} z_{ij} = n-1$) if $r_j = 1$, no link incident to component c_j is selected (i.e., $\sum_{c_i \in C} z_{ij} = 0$) if $r_j = 0$. Constraints (1.c) ensure that all links are not-oriented.

The clique cost f_s related to a feasible combination of train routing assignments s is computed as follows:

$$f_s = \sum_{c_i \in C} u_i r_i + \sum_{l_{ij} \in L} w_{ij} z_{ij} \quad (2)$$

where u_i is a non-negative cost associated with component c_i and w_{ij} is a non-negative cost associated with link l_{ij} . The costs u_i and w_{ij} indicate the undesirability of selecting the particular component c_i (i.e., a particular routing assignment

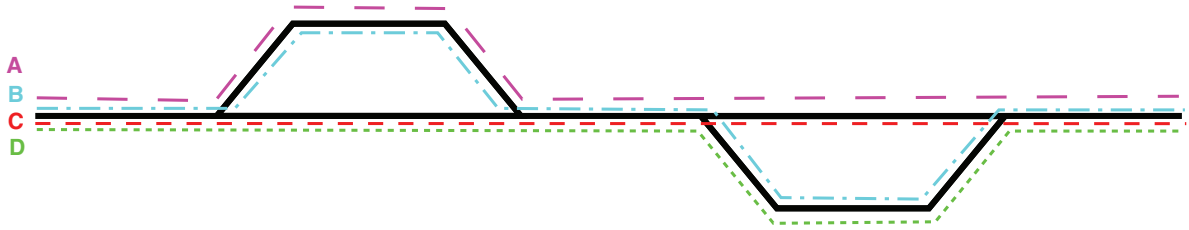


Fig. 3. A toy network.

Table 1
Example of constructing an rtTRSP solution.

S	T_1	T_2	T_3	T_4
s_1	c₁	c₅	c₇	c₁₀
s_2	c ₁	c₆	c₈	c ₁₀
s_3	c₂	c ₅	c ₈	c₁₁
	c₁, c₂	c₅, c₆	c₇, c₈	c₁₀, c₁₁

for a train) respectively for itself or in combination with c_j . We will investigate different strategies to compute the costs associated with components and links in Section 5.

The rtTRSP corresponds to the following formulation:

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

where q_s is a binary variable associated to a clique $s \in \Gamma$ and stating whether a clique s is part of an rtTRSP solution S ($q_s = 1$) or not ($q_s = 0$). We recall that p is the maximum cardinality of the subset of routings selected for each train. The constraints ensure that there are exactly m different feasible combinations of train routing assignments in S , where $m \geq p$ in order to cover multiple selections of the same component in different $s \in S$. An rtTRSP solution S thus corresponds to a set of m cliques in the construction graph G .

4.2.1. An example of rtTRSP

We present an illustrative example with four trains (t_1, t_2, t_3, t_4) and four routings (A, B, C and D) available in the network (see Fig. 3). However, there is a limited set of routing alternatives for each train: t_1 can use all four routings, t_2 and t_4 only routings C and D and t_3 routings B, C, D.

The construction graph G associated with the example has $\{c_1, c_2, c_3, c_4\} \in T_1$, $\{c_5, c_6\} \in T_2$, $\{c_7, c_8, c_9\} \in T_3$, $\{c_{10}, c_{11}\} \in T_4$. All train routing assignments are coherent, thus $|L| = 44$.

Table 1 presents an rtTRSP solution (rows 2, 3 and 4) and the relative input for the rtRTMP (row 5) for $m = 3$ and $p = 2$. For instance, a feasible combination of train routing assignments is shown in row 2 (e.g., $s_1 = \{c_1, c_5, c_7, c_{10}\}$). Each column shows the component chosen for the relative train in each of the m feasible combinations of train routing assignments among the $|\Gamma| = 48$ possible assignments. Regarding the input for the rtRTMP, the routing subset selected from S for t_1 contains only the routings A and B (corresponding to the selected components c_1 and c_2), thus limiting the number of routing alternatives to be considered by the rtRTMP algorithm. A similar reasoning is applied to the other three trains.

Fig. 4 depicts the three feasible combinations of train routing assignments of Table 1. The selected components of each feasible routing assignment for each train (e.g., s_1 is shown in the construction graph G on the left of Fig. 4) are colored in black, while the other components are colored in grey. Only the links between the selected components are depicted.

5. The rtTRSP cost definition

This section describes the methods developed to define the strategies to compute the costs associated with components and links in the construction graph. These costs are based on the potential delay computed for each train routing assignment. We define the potential delays based on train routing and scheduling decisions. These two types of decisions contribute independently on the definition of the potential delays.

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

The potential delay due to train scheduling is based on the train ordering decisions between pairs of coherent train routing assignments. For each pair of coherent train routing assignments c_i and c_j , the cost w_{ij} of link l_{ij} represents the

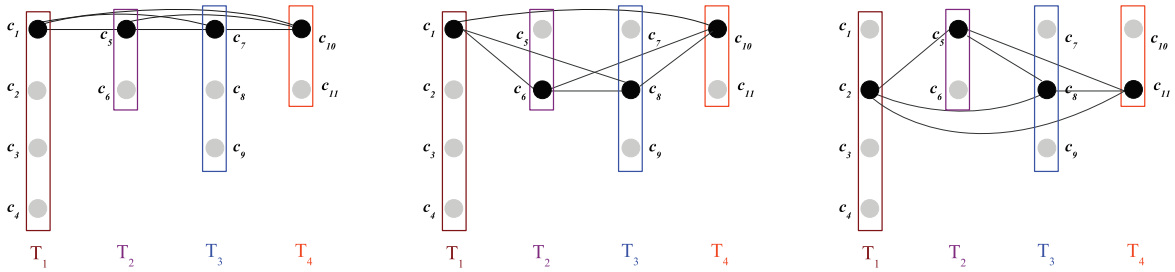


Fig. 4. Feasible combination of train routing assignments in G for s_1 (left), s_2 (centre), s_3 (right).

potential delay due to the corresponding train ordering decision. This potential delay is computed based on the consecutive delays measured when the train using c_i follows the train using c_j or vice versa. Specifically, we distinguish between two cases: the two trains travel in the same direction or in 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. On this train conflicting request, we order the two trains such that the consecutive delay is minimum. The latter delay represents the potential delay due to this scheduling decision.

If the two trains travel in the same direction, the following two steps are required for the computation of the potential delay: (i) a selection of the set of common resources between the two train routing assignments, (ii) an estimation of the potential delay on the selected set of common resources.

For step (i), we conceive the following alternative strategies based on the train conflicting requests between the two train routing assignments:

- *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*, computed as follows. For each conflicting request, we first identify the resources k^i and k^j requiring the maximum utilization time for train routing assignments c_i and c_j respectively. If $k^i = k^j$, this resource is included in the set. Otherwise, $k^i \neq k^j$ and the resource with the minimum utilization time between k^i and k^j is included in the set;
- *Max* – the set contains the resource k in each conflicting request with the *maximum maximum utilization time*. This strategy differs from the previous one only if $k^i \neq k^j$, in which case the resource with the maximum utilization time between k^i and k^j is included in the set.

For step (ii), we use the set of resources selected in step (i) and estimate the potential delay on those resources for each pair of coherent train routing assignments by one among 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, 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 as the maximum among the potential delays for all selected resources.

For $l_{ij} \in L$, let \tilde{w}_{ij} be the potential delay due to train ordering decisions computed as described above, and let k_{ij} be a resource where the potential delay is estimated for the components c_i and c_j . If $\tilde{w}_{ij} > 0$, we set $w_{ij} = \tilde{w}_{ij}$ in the construction graph G . Otherwise, we distinguish the two cases in which c_i and c_j do or do not have common resources. If c_i and c_j have at least one common resource, some unpredicted train conflicting requests could arise when solving the rRTMP. We therefore set $w_{ij} = 1$. If the train routing assignments do not share resources, there will never be a conflicting request and $w_{ij} = 0$.

In Section 7, we will combine the three strategies developed for step (i) with the three strategies developed for step (ii) and select the best rTRSP cost definition through experimental analysis.

5.1. An example of the rTRSP cost definition

This section shows how to define rTRSP costs for the illustrative example of Section 4.2.1 by means of the two-step procedure of Section 5. Specifically, we compute the three strategies of steps (i) and (ii) for the two trains (t_1 and t_3) traversing the network in the same direction. Fig. 5 shows the routing assignments of the two trains ($c_1 \in T_1$ and $c_7 \in T_3$), that would generate two conflicting requests: the first one on the resources res_1 , res_2 and res_3 , the second one on res_5 . For

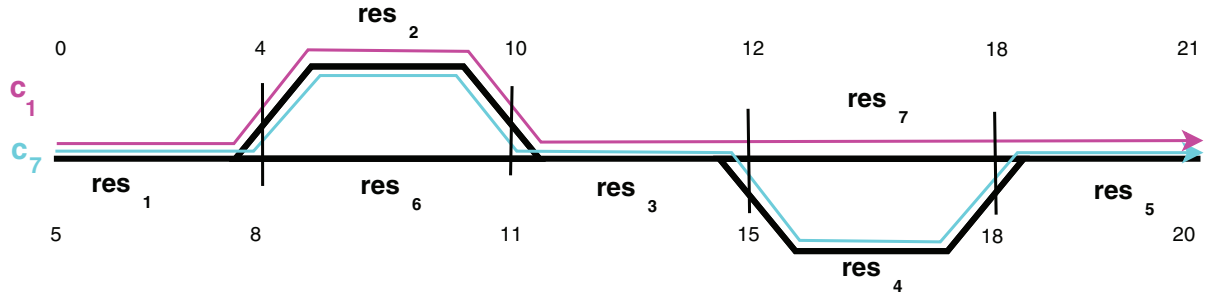


Fig. 5. Example of two train routing assignments for the example of Section 4.2.1.

the sake of clarity, we assume in this example that all reservation times and clearing times are null. Fig. 5 shows the start of the utilization time of c_1 (c_7) on the top (bottom) left of each resource.

The cost w_{17} of the link l_{17} connecting the components c_1 and c_7 is computed as follows. Step (i) requires the computation of the following quantities with respect to the three proposed strategies:

- *All* – the set of the selected resources is formed by res_1 , res_2 , res_3 and res_5 .
- *Min* – we analyze each of the two conflicting requests. For the first one, the resource with the maximum utilization time is: res_2 for c_1 ($k^{c_1} = res_2$, with a duration of $10 - 4 = 6$); res_3 for c_7 ($k^{c_7} = res_3$, with a duration of $15 - 11 = 4$). Being $k^{c_1} \neq k^{c_7}$, we select the resource with the minimum utilization time, i.e., $k = res_3$. For the second conflicting request, the only conflicting resource is res_5 , therefore $k = res_5$. The set of the selected resources is formed by res_3 and res_5 .
- *Max* – as for the Min strategy, $k^{c_1} = res_2$ and $k^{c_7} = res_3$. In addition, $k_{c_1} \neq k_{c_7}$. Between the two train conflicting resources we select the one with the maximum utilization time, i.e., $k = res_2$. The set of selected resources is formed by res_2 and res_5 .

For the explanation of step (ii), we consider the set of selected resources computed by the Min strategy. The potential delay of l_{17} can be computed by one of the three proposed strategies:

- *Train* – for t_1 , the consecutive delay generated when waiting t_3 is 5 ($15 - 10$) on res_3 and 2 ($20 - 18$) on res_5 . The potential delay of t_1 is thus 5 on res_3 . For t_3 , the consecutive delay generated when waiting t_1 is 1 ($12 - 11$) on res_3 and 3 ($21 - 18$) on res_5 . The potential delay of t_3 is thus 3 on res_5 . The potential delay of the pair is the minimum, i.e., $\tilde{w}_{17} = 3$ and $k_{17} = res_5$.
- *ResMin* – for res_3 , t_1 would have a consecutive delay of 5, t_3 of 1. The potential delay on res_3 is 5. For res_5 , t_1 would have a consecutive delay of 2, t_3 of 3. The potential delay on res_5 is 3. The potential delay of the pair is $\tilde{w}_{17} = 3$ and $k_{17} = res_5$.
- *ResMax* – for res_3 , t_1 would have a consecutive delay of 5, t_3 of 1. The potential delay on res_3 is 1. For res_5 , t_1 would have a consecutive delay of 2, t_3 of 3. The potential delay on res_5 is 2. The potential delay of the pair is $\tilde{w}_{17} = 2$ and $k_{17} = res_5$.

In all three cases of step (ii) $\tilde{w}_{17} > 0$, thus $w_{17} = \tilde{w}_{17}$. In the whole example, all train routing assignments share at least one resource, therefore each link with $\tilde{w}_{ij} = 0$ has value $w_{ij} = 1$ in the construction graph.

6. The ACO-rtTRSP algorithm

This section presents a meta-heuristic algorithm which tackles the rtTRSP. The proposed algorithm is inspired by an ACO algorithm originally developed for the maximum clique problem (Solnon and Bridge, 2006a). The reasons behind the choice of the ACO meta-heuristic are the two-fold: 1) the rtTRSP is similar to a subset selection problem and the ACO algorithm performs well for this type of problem when compared with other heuristic approaches (Solnon and Bridge, 2006b); 2) ACO fits particularly well the real-time and combinatorial nature of the rtTRSP and computes multiple good quality solutions in a very short computation time.

ACO is a meta-heuristic inspired by the foraging behavior of ant colonies. Solutions are incrementally constructed by each of the $nAnts$ ants of a colony. At each step of the solution construction process, an ant a selects a new solution component probabilistically using the random proportional rule, which is based on *pheromone trails* and *heuristic information* (i.e., the colony's shared knowledge and a greedy measure on the quality of a component which may be added to the current partial solution s_a). Once all the ants of the colony have built a feasible solution, the best one is stored and the pheromone trails are updated accordingly. This process is repeated iteratively until the available computation time has elapsed.

We now describe the ACO algorithm developed for the rtTRSP, in which we apply the following slight abuse of terminology in order to be consistent with the literature on ACO algorithms. An ACO-rtTRSP solution is a set of n coherent train routing assignments, while an rtTRSP solution is derived from the best m ACO-rtTRSP solutions as follows. The routing subset of each train includes the timetable routing plus the $p - 1$ routing alternatives belonging to the best ACO-rtTRSP solution.

Since the routing subset must not include duplicates, we replace the routings which appear more than once in these best solutions with randomly chosen ones.

The ACO algorithm works on the construction graph G as follows. For each ant a , the first component $c_i \in C$ is randomly selected and is inserted in s_a . The set of *candidates* among which the ant can choose the next component $c_j \in C$ includes all the components linked to c_i , i.e., all coherent train routing assignments. The component c_j is chosen with a probability computed via the random proportional rule, that is based on the following pheromone trails and heuristic information. The value of the heuristic information associated to each $l_{ij} \in L$ is $\eta(l_{ij}) = \frac{1}{1+w_{ij}+u_j+IN_{ij}}$, stating the desirability of selecting c_j when considering c_i and IN_{ij} . IN_{ij} is a counter of the number of links which connect the pairs of components (c_h, c_z) for which $k_{hz} = k_{ij}$ (the resources where the potential delays is estimated on both links is the same) in the current partial solution s_a . We remark that the higher is IN_{ij} , the smaller is $\eta(l_{ij})$. The use of IN_{ij} in the formula favors the use of links with no new conflicting requests on the resources having already a high number of conflicting requests in s_a .

After each addition of a component c_j in the partial solution, the set of candidates is updated by removing both c_j and all components $c_h \in C: \exists l_{jh} \in L$, i.e., all train routing assignments not coherent with the train routing assignment corresponding to c_j . The solution construction process terminates when the set of candidates is empty. A complete solution s_a is feasible only if it includes n components, where n is the number of trains in the network.

Among the $nAnts$ solutions found in the current iteration, let $BestS$ be the feasible solution with minimum cost. After each iteration, the following local search is applied to $BestS$ to try to improve it: a solution component c_d in $BestS$ is replaced by another component c_e such that $c_d, c_e \in T_t$ and c_e has a link with each of the other components in $BestS$, and it minimizes the clique cost $CostBestS$ of $BestS$. To select the component c_d , we consider three possible local search strategies (*localSearchStr*):

- *Random* – a random selection;
- *Cost* – the selection of the component having $\max\{u_d + \sum_{b \in BestS} w_{db}\}$, i.e., the one impacting most on $CostBestS$;
- *k* – the selection of the component that maximizes the number of links l_{db} such that $k_{db} = \hat{k}$, with \hat{k} being the resource having the highest number of conflicting requests in the clique.

At the end of each iteration, an update of the pheromone trails is required. The ACO-rtTRSP algorithm is based on the MAX-MIN Ant System (Stützle and Hoos, 2000), in which upper and lower bounds are imposed on the pheromone trails. According to Solnon and Bridge (2006a), we use $\tau_{Min} = 0.01$ and $\tau_{Max} = 6$. During the update of the pheromone trails, some pheromone evaporates from all links of G and some additional pheromone is deposited on the links belonging to $BestS$, following the *clique pheromone strategy* (Solnon and Bridge, 2006a).

The m best ACO-rtTRSP solutions are stored in an ordered solution set *PoolBestSol* as follows. The position of a solution in *PoolBestSol* can be based on two ordering criteria: 1) its clique cost, being the best solution the one with minimum clique cost; 2) a lexicographic bi-objective function: the first objective is the minimization of the clique cost, the second is the minimization of the number of conflicting requests between train routing assignments on a resource, being the best solution the one with minimum number.

The algorithm terminates the search when a given time limit of computation $time_{max}$ seconds is reached or when all m solutions have a null cost. The pseudo-code of the ACO-rtTRSP algorithm is reported in Fig. 6.

The ACO-rtTRSP parameters are next summarized:

- α , the influence of pheromone trails in the random proportional rule;
- β , the influence of heuristic information in the random proportional rule;
- $nAnts$, the number of ants in the colony;
- ρ , the pheromone evaporation rate;
- *localSearchStr*, the strategy used to select the component on which the local search is performed;
- *count*, determining if the counter IN is used (T, True) or not (F, False) in the computation of heuristic information;
- *biObj*, determining if *PoolBestSol* is generated by using the ordering criterion 1 (F) or 2 (T).

The next section will describe how we set these parameters based on a state-of-the-art automatic tuning procedure.

7. Computational experiments

This section presents the computational results on the rtTRSP and rtRTMP. The rtTRSP is solved by the ACO-rtTRSP algorithm of Section 6 or by a random approach, while the rtRTMP is solved by exact and heuristic approaches of Pellegrini et al. (2014, 2015). All tests are performed with a Quad-Core Intel Xeon E5 3.7 GHz processor with 32GB RAM, under OS X 10.10.3.

The computational experiments have been performed in a laboratory environment for two French case studies based on real-world data. The first case study concerns the 27-km-long railway line around the city of Rouen, shown in Fig. 7, with 190 track-circuits, 189 block sections, and 11,347 routings. This line presents interesting practical aspects: there are several intermediate stations, each one with up to six platforms; a part of the line is equipped to be traversed on both directions.

The second case study is the 12-km-long Lille station area shown in Fig. 8, with 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.

ACO-rtTRSP ALGORITHM

Input: G , α , β , $nAnts$, ρ , $localSearchStr$, $count$, $biObj$, $time_{max}$, n

Set pheromone trails $\tau(l_{ij}) = \tau_{Max} \quad \forall l_{ij} \in L$

while ($time \leq time_{max}$) **do**

$CostBestS \leftarrow \text{MAX INT}$

for ($a = 0$; $a < nAnts$; $a++$) **do**

Randomly choose $c_i \in C$

$s_a \leftarrow \{c_i\}$

$Candidates \leftarrow \{c_j \in C : \exists l_{ij} \in L\}$

while ($Candidates \neq \emptyset$) **do**

choose $c_j \in C$ with probability $\frac{\tau(c_j, s_a)^\alpha \eta(c_j, s_a)^\beta}{\sum_{c_h \in Candidates} \tau(c_h, s_a)^\alpha \eta(c_h, s_a)^\beta}$

$s_a \leftarrow s_a \cup \{c_j\} \cup \{l_{ij} : c_i \text{ in } s_a\}$

$Candidates \leftarrow \text{Remove } c_j \cup \{c_h \in C : \nexists l_{jh} \in L\}$

end while

if ($f_{s_a} < CostBestS$ & s_a feasible) **then**

$BestS \leftarrow s_a$

$CostBestS \leftarrow f_{s_a}$

end if

end for

Apply $LocalSearch(localSearchStr)$ to $BestS$

Update pheromone trails $\tau(l_{ij})$ in G by using *clique pheromone strategy*

Store $BestS$ in the ordered solution set $PoolBestSol$

end while

Output: Return $s_a \in PoolBestSol$ with $a = 1, \dots, m$

Fig. 6. Pseudo-code of the ACO-rtTRSP algorithm.

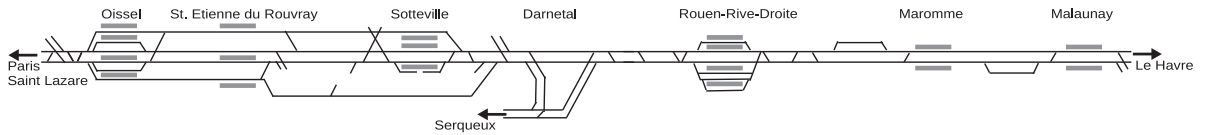


Fig. 7. Rouen network.

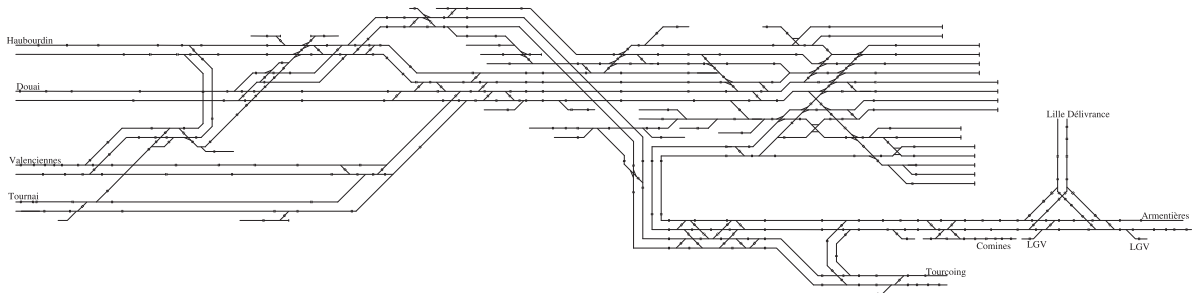


Fig. 8. Lille network.

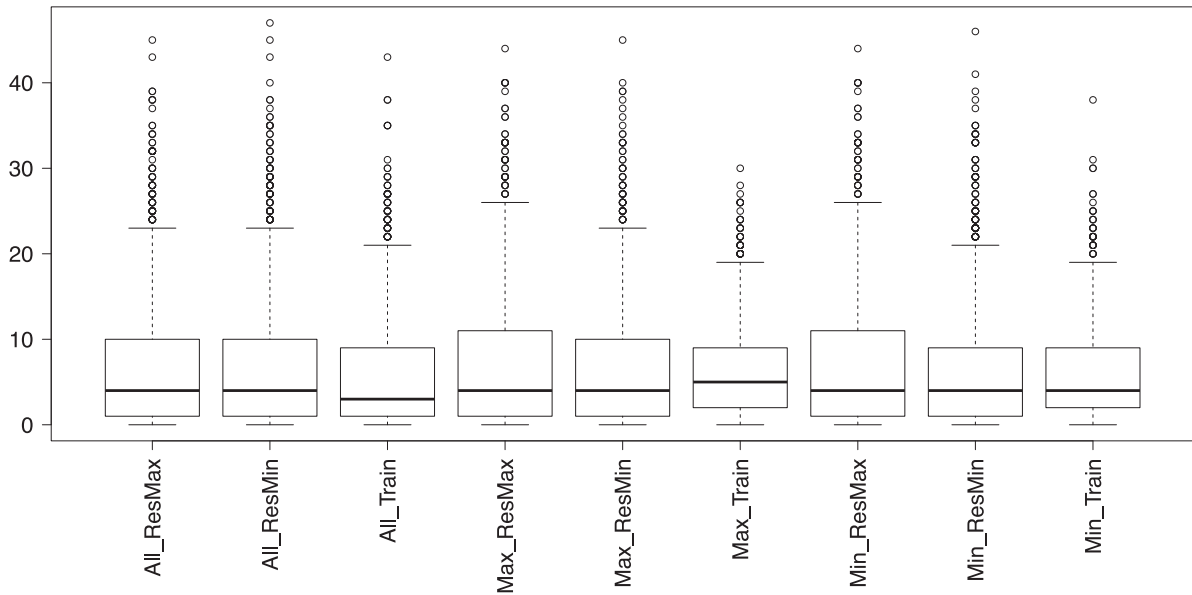


Fig. 9. Boxplots of the difference in absolute value between the rtTRSP and rtRTMP rankings.

A typical daily timetable has 186 trains for the Rouen case study and 509 for the Lille one. In our computational experiments, we deal with 21 scenarios for the Rouen case study and 15 scenarios for the Lille case study that are generated from a typical daily timetable as follows. In each scenario, 20% of the trains, randomly selected, are affected by a random delay between 5 and 15 min applied at their entry point. For each scenario, we generate 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 traffic peak-hours (i.e., in the time periods 7:30–9:00 and 18:30–20:00). In total, we obtain 210 rtRTMP instances for the Rouen and 150 for Lille case study. In the Rouen instances, each train can have a maximum of 192 routing alternatives and the average number of trains is 13. In the construction graph G , the average number of components and links is $|C| = 597$ and $|L| = 152,441$. In the Lille instances, each train can have a maximum of 100 routing alternatives and the average number of trains is 40. In the construction graph G , the average number of components and links is $|C| = 3881$ and $|L| = 6,902,867$ for the Lille instances.

In the remaining of this section, we present an assessment of the strategies to select the rtTRSP costs (Section 7.1), of the ACO-rtTRSP parameters (Section 7.2), of the ACO-rtTRSP algorithm (Section 7.3), of the algorithms to solve the rtTRSP and rtRTMP (Section 7.4).

7.1. Selection of the rtTRSP costs

This section reports a computational analysis on 30 Rouen instances dedicated to the selection of the best rtTRSP cost definition of Section 5. For each Rouen instance and for each cost definition, we generate 50 different rtTRSP solutions by using the ACO-rtTRSP algorithm for $m = 1$. In total, we have 1500 rtTRSP solutions for each cost definition. Starting from each of the 1500 rtTRSP solutions, we compute the optimal solution for the corresponding rtRTMP.

The 1500 rtTRSP and rtRTMP assessments are compared as follows. For each Rouen instance, we compute two rankings of the 50 solutions: 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 definition that presents the minimum absolute difference with respect to the rtRTMP assessment. This difference is obtained through the Wilcoxon rank-sum test with a confidence level of 0.95. We observe that the two rankings are significantly different for all nine cost definitions.

Fig. 9 shows a boxplot comparison of the rtTRSP cost definitions of Section 5 in terms of the of the absolute value of the difference between the rtTRSP and the rtRTMP rankings. Each rtTRSP cost definition is named according to the scheme $\langle set \rangle_ \langle estimation \rangle$. This scheme reflects the two steps of Section 5 used to estimate the potential delays. Specifically, $\langle set \rangle$ refers to the strategy used for step (i), i.e., the criterion (All, Max, Min) to select the set of common resources to be considered in the estimation; $\langle estimation \rangle$ refers to the strategy used for step (ii), i.e., the strategy (Train, ResMin, ResMax) to estimate the potential delay on the selected common resources. The thick horizontal line in the boxplot represents the median of the distribution, the extremes of the boxplot are the first and third quartiles, the dots are the outliers, and the whiskers show the smallest and the largest non-outliers in the data-set.

Table 2
Parameter values considered during the tuning.

α	β	$nAnts$	ρ	$localSearchStr$	$Count$	$biObj$
1, 2, 3, 4	2, 3, 4, 5	25, 50, 100, 150 , 200	0.02, 0.05 , 0.2, 0.7	Cost , random, k	T, F	T, F

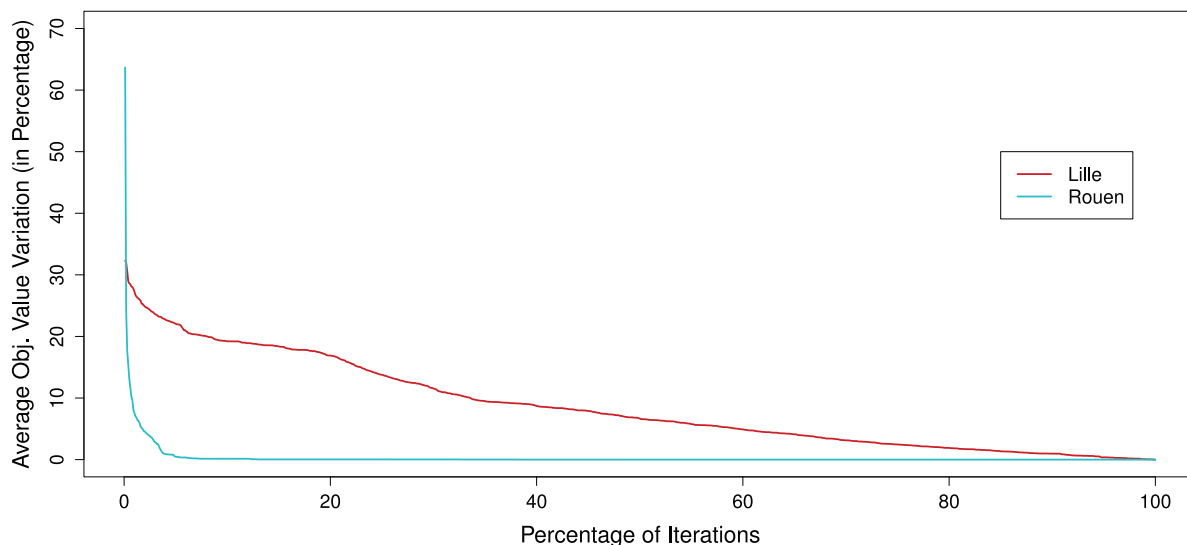


Fig. 10. ACO-rtTRSP best solution value when varying the number of iterations.

In addition to the boxplot comparison, we consider the value of the *pseudo-median* (i.e., an estimation of the population's location parameter that is closely related to its median) of the distributions: it gives an indication of what would be the median of the distribution of the absolute difference of the two solution rankings. We use this indication in combination with the boxplots in order to establish which is the best rtTRSP cost definition.

The cost definitions sharing the best minimum value (6) of the pseudo-median are: *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. In order to select one among the three best cost definitions, we look at Fig. 9. The best cost definition appears to be *All_Train*, since it has a median of 3 against 4 of *Min_Train* and 5 of *Max_Train* and is the best until the 75th percentile of the distribution, where the three best cost definitions are equivalent with an absolute difference of 9. In the next subsections, we consider *All_Train* set as the rtTRSP cost definition.

7.2. Tuning of the ACO-rtTRSP parameters

This section reports a computational analysis on 30 Rouen instances, different from the ones tested in Section 7.1, dedicated to the selection of the best combination of the ACO-rtTRSP parameters. The rtTRSP related to each Rouen instance is solved by the ACO-rtTRSP algorithm with a computation time limit of 30 s.

The ACO-rtTRSP parameters are tuned by using IRACE (Iterated Racing for Automatic Algorithm Configuration, López-Ibáñez et al., 2011), in which the best parameter setting is selected through an iterated racing procedure. We set to 35,000 the maximum number of runs to be performed in IRACE. Table 2 presents the settings tested for each parameter. We highlight in bold the settings selected.

7.3. Convergence of the ACO-rtTRSP algorithm

This section gives information on the rapidity of the ACO-rtTRSP algorithm to converge to the best-known solution. We present average results on the remaining 150 Rouen instances and on the 150 Lille instances when varying the percentage of iterations performed by the algorithm in the given time limit of computation. We use the percentage of iterations because the time required by the algorithm to perform each iteration varies from instance to instance, and from test case to test case. Fig. 10 shows the average variation of the objective function value computed as follows: (objective function value) - (best objective function value)/(best objective function value), expressed in percentage.

Comparing the results obtained in Fig. 10 for the two test cases, the average convergence of the algorithm is significantly faster for the Rouen instances, since $|C|$ and $|L|$ are smaller in the corresponding construction graph G and the algorithm

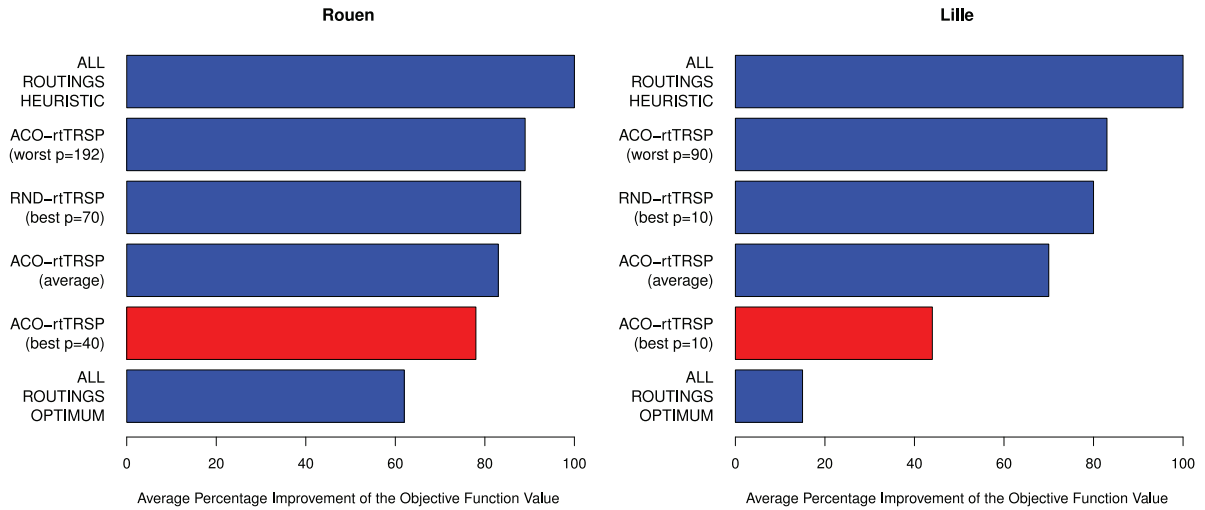


Fig. 11. Barplots of the average percentage improvement with respect to the benchmark value.

performs a high number of iterations in the given computation time. For the Lille instances, the algorithm takes a longer computation time per iteration and sometimes uses all the given time in order to find the best-known solution.

7.4. Comparison between algorithms

This section presents a computational comparison of rtTRSP and rtRTMP algorithms on the 150 Rouen instances and on the 150 Lille instances, also used in Section 7.3. Specifically, we use the ACO-rtTRSP algorithm with the best value for each parameter of Section 7.2, the exact and heuristic rtRTMP approaches of Pellegrini et al. (2014, 2015). The two rtRTMP algorithms require a starting routing alternative for each train, that is used to compute an initial upper bound for the rtRTMP, and they may deal with a set of routing alternatives. In this experiments, the rtTRSP and rtRTMP algorithms are used to generate a working timetable via the following approaches:

1. **ALL ROUTINGS HEURISTIC:** All routing alternatives are given in input to the best heuristic algorithm in Pellegrini et al. (2015) that returns the best rtRTMP solution found within 180 s. The timetable routing is set as the starting routing alternative for each train;
2. **RND-rtTRSP:** An rtTRSP solution with p routing alternatives for each train is generated by fixing the starting routing alternative for each train as its timetable routing and by selecting randomly the remaining $p - 1$ routing alternatives. This rtTRSP solution is given in input to the same algorithm of approach 1 that returns the best rtRTMP solution found within 180 s;
3. **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 s. This rtTRSP solution is given in input to the same algorithm of approach 1 that returns the best rtRTMP solution found within 150 s. The best ACO-rtTRSP routing for each train is set as its starting routing alternative;
4. **ALL ROUTINGS OPTIMUM:** All routing alternatives of each train are given in input to the exact approach in Pellegrini et al. (2014, 2015) that returns the optimal rtRTMP solution. The timetable routing is set as the starting routing alternative for each train.

For the ALL ROUTINGS OPTIMUM approach, we used the exact approach in Pellegrini et al. (2014, 2015) with no time limit of computation. All the 150 Rouen instances and 91/150 Lille instances were solved to optimality between around 1 and 3 h of computation. Regarding the other 59 Lille instances, the algorithm was interrupted due to a lack of memory and returned the best solution found.

Fig. 11 shows a barplot comparison between the above introduced approaches for the Rouen and Lille case studies. We consider the best average rtRTMP solution found by the ALL ROUTINGS HEURISTIC approach as the benchmark value, since this is our reference state-of-the-art approach to solve the rtRTMP with a computation time compatible with real-time operations. Each bar gives the percentage improvement of the average objective function value with respect to the benchmark value. For the RND-rtTRSP approach we consider the best values of p , i.e., 70 (10) for the Rouen (Lille) case study. For the ACO-rtTRSP approach we consider the cases: (i) the best values of p , i.e., 40 (10) for the Rouen (Lille) case study; (ii) the worst values of p , i.e., 192 (90) for the Rouen (Lille) case study; (iii) the average results obtained when varying the values of p . The best and worst p values are obtained for the 300 instances of the two case studies, when varying p in the range [10, 20, 30, ..., 170, 180, 190, 192] ([10, 20, 30, ..., 80, 90, 100]) for the Rouen (Lille) instances. Tables 3 and 4 will give further information on the results with different p Values.

Table 3

Comparison between ACO-rtTRSP and RND-rtTRSP on the Rouen results.

p	# Opt. sol. ACO-rtTRSP (/150)	# Opt. sol. RND-rtTRSP (/150)	Obj. value ACO-rtTRSP (s)	Obj. value RND-rtTRSP (s)	Comp. time ACO-rtTRSP (s)	Comp. time RND-rtTRSP (s)
10	88	28	178.1	220.4	62.4	54.7
20	90	53	173.3	203.1	83.7	82.1
30	91	70	176.0	217.6	101.1	101.8
40	100	67	168.3	196.9	113.4	116.1
50	98	77	173.0	216.4	119.9	118.6
60	100	76	170.9	223.3	126.8	127.2
70	99	80	173.0	189.2	127.0	130.1
80	99	90	176.2	201.1	133.6	128.4
90	99	85	168.5	202.3	135.9	137.8
100	101	90	181.0	205.4	137.5	139.9
110	98	79	182.6	226.3	136.4	143.0
120	95	93	181.8	220.1	138.4	138.4
130	96	93	178.9	206.0	138.8	137.0
140	97	93	190.1	203.8	140.3	137.4
150	98	92	178.9	223.9	138.3	143.8
160	97	94	169.2	205.4	138.9	139.7
170	96	88	191.3	204.0	141.4	141.8
180	99	90	193.1	246.0	143.0	141.5
190	96	87	183.4	221.8	141.2	144.2
192	95	91	193.2	216.1	143.2	143.8

Table 4

Comparison between ACO-rtTRSP and RND-rtTRSP on the Lille results.

p	# Opt. sol. ACO-rtTRSP (/91)	# Opt. sol. RND-rtTRSP (/91)	Obj. value ACO-rtTRSP (s)	Obj. value RND-rtTRSP (s)	Comp. time ACO-rtTRSP (s)	Comp. time RND-rtTRSP (s)
10	6	7	615.1	1111.0	160.4	162.8
20	10	6	731.9	1280.6	175.9	180.0
30	6	3	879.5	1375.5	180.0	180.0
40	6	3	953.5	1388.4	180.0	180.0
50	8	1	1008.4	1398.5	180.0	180.0
60	7	3	1047.0	1428.1	180.0	180.0
70	5	0	1062.4	1407.7	180.0	180.0
80	5	2	1077.1	1424.2	180.0	180.0
90	6	2	1145.1	1402.4	180.0	180.0
100	6	0	1098.7	1383.0	180.0	180.0

The results of Fig. 11 illustrate the improvement achievable when using the rtTRSP solutions as input for the rtRTMP heuristic RECIFE-MILP (Pellegrini et al., 2015). An improvement of around 12% (20%) is already obtained for the Rouen (Lille) instances when using random rtTRSP solutions, since the heuristic improves on average its performance on the reduced rtRTMP. A further improvement of around 10% (36%) is obtained for the Rouen (Lille) instances when using the ACO-rtTRSP algorithm for the best p Values rather than the random approach, since the heuristic works on the reduced rtRTMP with an optimized subset of routing alternatives for each train. However, a potential improvement of around 16% (29%) can still be achieved for the Rouen (Lille) instances when looking at ALL ROUTINGS OPTIMUM. The potential improvement is larger for the Lille instances, since the central part of the Lille station is densely used in both traffic directions, many trains share the same track-circuits and the primary delay quickly propagates.

The results obtained by the ACO-rtTRSP algorithm for the worst p values and the average results obtained by the ACO-rtTRSP algorithm when varying the values of p show that even a *wrong* choice of the parameter p would still give an improvement over the ALL ROUTINGS HEURISTIC approach.

Tables 3 and 4 report a detailed comparison between the ACO-rtTRSP and RND-rtTRSP approaches on the 150 Rouen and Lille instances, respectively. The comparison is reported, on average, for each value of p in terms of the number of rtRTMP solutions that are optimal when considering all train routing alternatives, the objective function value and the computation time. For each case study and rtTRSP approach, the best average value of the objective function and the corresponding p value are highlighted in bold.

From the results of Tables 3 and 4 we have the following observations:

- The best p Values are small values (especially for the ACO-rtTRSP approach): in the available computation time, RECIFE-MILP does not manage to improve its performance when there is a too large number of routing alternatives for each train, even if the optimal routing alternatives are more likely to be considered for large p values;

- The ACO-rtTRSP approach is better than the RND-rtTRSP approach for any row of Tables 3 and 4. This result highlights the importance of optimizing the selection of the routing subset for each train.
- Looking at the case when all routing alternatives are considered (i.e., the last row of Tables 3 and 4), the ACO-rtTRSP approach is better than the RND-rtTRSP approach, and it properly chooses the starting routing alternative for each train. For the Rouen (Lille) case study, the starting rtRTMP upper bound is 233 (1493) when using the ACO-rtTRSP approach, while 1469 (1550) when using the RND-rtTRSP approach. Another added value of the ACO-rtTRSP approach for better solving the rtRTMP is therefore the selection of a good quality starting routing alternative for each train, that results to be better than the timetable routing in the studied perturbed scenarios and network configurations.
- The problem of finding a good quality rtRTMP solution in a short time becomes particularly challenging for large values of p , since the number of rtRTMP variables significantly increases. This trend motivates the focus of this paper in searching for the best selection of alternative routings for each train.

8. Conclusions and future research

This paper investigates systematically the improvement achievable in the solution of the rtRTMP when optimizing the selection of the starting routing alternative and the alternative routing subset for each train. This problem is named the rtTRSP and is important for the following reasons: 1) the dispatcher needs indications on how to re-optimize the routing of trains during disturbed operations; 2) the identification of the potential best routing alternatives for each train can improve the existing rtRTMP algorithms. Due to the real-time and combinatorial characteristic of the rtTRSP, this problem is modeled as an integer linear programming formulation and efficiently solved by an ACO algorithm in a short computation time. The computational results on two real-world case studies of the French railways confirm the advantage of using the rtTRSP solutions, as input data for the solution of the rtRTMP, compared to a state-of-the-art rtRTMP algorithm. The largest advantage (up to 22% for the Rouen case study and up to 56% for the Lille case study) is obtained when solving the rtTRSP seeking a small number of suitable alternative routings for each train.

Future research shall be dedicated to further improve the quality of the rtRTMP solutions. This can be achieved by developing better quality combinations between rtRTMP and rtTRSP approaches, rtRTMP heuristics, rtTRSP cost definitions and ACO algorithmic settings. Other research directions shall be dedicated to the generalization of the ACO-rtRTMP approach to different objective functions and constraints, that may be more appropriate when solving the rtRTMP for other types of traffic flows, disturbances and infrastructure configurations.

Acknowledgment

This work was partially supported by the “Ambassade de France en Italie” with a three-month MAE mobility scholarship for Ph.D. candidates. We would like to thank the anonymous reviewers for their helpful and constructive remarks.

Appendix

In this Appendix, we detail the MILP formulation for the rtRTMP, introduced in Pellegrini et al. (2014, 2015). In the MILP formulation, we use the following notation:

- tc_0 \equiv entry location in the network considered;
- tc_∞ \equiv exit location from the network considered;
- T \equiv set of trains;
- ty_t \equiv type (rolling stock characteristics) of train t ;
- $init_t$ \equiv earliest time at which train t can be operated, including the entrance delay if any;
- $exit_t$ \equiv maximum between the scheduled exit time of train t from the network and the earliest exit time computed by considering the given $init_t$, the timetable routing and the scheduled stops;
- $i(t', t)$ \equiv indicator function: 1 if t' and t use the same rolling stock and t results from the turnaround or join or split of t' , 0 otherwise;
- $ms_{t', t}$ \equiv minimum separation time between the arrival and departure of two trains using the same rolling stock ($i(t', t) = 1$);
- R_t \equiv set of alternative routings for train t ;
- TC^r \equiv set of track-circuits composing routing r ;
- $OTC_{ty, r, tc}$ \equiv set of track-circuits occupied by a train t of type ty along r if the head of train t is at the end of track-circuit tc (\emptyset if t is shorter than tc);
- $TC(tc, tc', r)$ \equiv set of track-circuits between tc and tc' along r ;
- $p_{r, tc}, S_{r, tc}$ \equiv track-circuits preceding and following tc along r ;
- $rt_{ty, r, tc}, ct_{ty, r, tc}$ \equiv travel and clearing time of tc along r for a train of type ty in absence of disturbances;
- $ref_{r, tc}$ \equiv reference track-circuit for the reservation of tc along r (it depends on the interlocking system);
- $bs_{r, tc}$ \equiv block-section including track-circuit tc along routing r ;
- for_{bs}, rel_{bs} \equiv formation time and release time for block-section bs ;

- $e(tc, r) \equiv$ indicator function: 1 if track-circuit tc belongs to either the first block-section or the last block-section of r , 0 otherwise;
- $S_t, TCS_{t,s} \equiv$ set of stations where t has a scheduled stop, and set of track-circuits that can be used by t for stopping at s ;
- $dw_{t,s}, d_{t,s} \equiv$ minimum dwell time and scheduled departure time for train t at station s ;
- $\hat{T}C_{t,t',tc} \equiv$ set of track-circuits tc' which may be used by both t and t' such that if t precedes (\prec) t' on tc , $t \prec t'$ on tc' , and so on (e.g., if the track-circuits follow each other on a straight track segment). $\hat{T}C_{t,t',tc} = \{tc\}$ if $tc \in \cup_{r \in R_t} TC^r \cap \cup_{r \in R_{t'}} TC^r$ and no implied precedence relation links tc to other track-circuits. $\hat{T}C_{t,t',tc} = \emptyset$ if $\exists tc' \in \cup_{r \in R_t} TC^r \cap \cup_{r \in R_{t'}} TC^r$ such that $tc \in \hat{T}C_{t,t',tc'}$, i.e., each track-circuit belongs to one and only one set $\hat{T}C_{t,t',tc}$;
- $M \equiv$ large constant.

The formulation uses non-negative *continuous variables*:

- for all triplets of $t \in T, r \in R_t$ and $tc \in TC^r$:
 - $o_{t,r,tc}$: time in which t starts the occupation of tc along r ,
 - $l_{t,r,tc}$: longer stay of t 's head on tc along r , due to dwell time and scheduling decisions (i.e., a consecutive delay);
- for all pairs of $t \in T$ and $tc \in \cup_{r \in R_t} TC^r$:
 - $sU_{t,tc}, eU_{t,tc}$: time in which t starts and ends tc utilization;
- for all $t \in T$:
 - D_t : consecutive delay suffered by train t when exiting the network.

In addition, the formulation 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 $tc \in \cup_{r \in R_t} TC^r \cap \cup_{r \in R_{t'}} TC^r$ such that $\hat{T}C_{t,t',tc} \neq \emptyset$:

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

The objective is the minimization of the total consecutive delays suffered by trains at their exit points:

$$\min \sum_{t \in T} D_t. \quad (4)$$

This objective function must be minimized while respecting the following constraints:

$$o_{t,r,s_{r,tc_0}} \geq \text{init}_t \quad \forall t \in T \quad r \in R_t \quad (5)$$

$$o_{t,r,tc} \leq Mx_{t,r} \quad \forall t \in T \quad r \in R_t \quad tc \in TC^r \quad (6)$$

$$o_{t,r,tc} = o_{t,r,p_{r,tc}} + l_{t,r,p_{r,tc}} + rt_{r,ty_t,p_{r,tc}} x_{t,r} \quad \forall t \in T \quad r \in R_t \quad tc \in TC^r \setminus \{tc_0\} \quad (7)$$

$$o_{t,r,s_{r,tc}} \geq d_{t,s} x_{t,r} \quad \forall t \in T \quad r \in R_t \quad s \in S_t \quad tc \in TCS_{t,s} \cap TC^r \quad (8)$$

$$l_{t,r,tc} \geq dw_{t,s} x_{t,r} \quad \forall t \in T \quad r \in R_t \quad s \in S_t \quad tc \in TCS_{t,s} \cap TC^r \quad (9)$$

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T \quad (10)$$

$$D_t \geq \sum_{r \in R_t} o_{t,r,tc_\infty} - \text{exit}_t \quad \forall t \in T \quad (11)$$

$$\sum_{\substack{r \in R_t, tc \in TC^r: \\ p_{r,tc} = tc_0}} o_{t,r,tc} \geq \sum_{\substack{r \in R_{t'}, tc \in TC^r: \\ s_{r,tc} = tc_\infty}} o_{t',r,tc} + (ms_{t',t} + rt_{r,ty_{t'},tc})x_{t',r} \quad \forall t, t' \in T : i(t', t) = 1 \quad (12)$$

$$\sum_{r \in R_t : tc \in TC^r} x_{t,r} = \sum_{r \in R_{t'} : tc \in TC^r} x_{t',r} \quad \forall t, t' \in T : i(t', t) = 1 \\ tc \in TC : \exists r \in R_t, tc = s_{r,tc_0} \quad (13)$$

$$\sum_{\substack{tc \in \cup_{r \in R_t} TC^r: \\ \exists r \in R_t, p_{r,tc} = tc_0}} sU_{t,tc} \leq \sum_{\substack{tc \in \cup_{r \in R_{t'}} TC^r: \\ \exists r \in R_{t'}, s_{r,tc} = tc_\infty}} eU_{t',tc} \quad \forall t, t' \in T : i(t', t) = 1 \quad (14)$$

$$sU_{t,tc} = \sum_{r \in R_t : tc \in TC^r} (o_{t,r,ref_{r,tc}} - for_{bs_{r,tc}} x_{t,r}) \quad (\nexists t' \in T : i(t', t) = 1) \vee (\forall r \in R_t : ref_{r,tc} \neq s_{r,tc_0}) \quad \forall t \in T \quad tc \in \cup_{r \in R_t} TC^r : \quad (15)$$

$$sU_{t,tc} \leq \sum_{r \in R_t : tc \in TC^r} (o_{t,r,ref_{r,tc}} - for_{bs_{r,tc}} x_{t,r}) \quad (\exists t' \in T : i(t', t) = 1) \wedge (\exists r \in R_t : ref_{r,tc} = s_{r,tc_0}) \quad \forall t \in T \quad tc \in \cup_{r \in R_t} TC^r : \quad (16)$$

$$eU_{t,tc} = \sum_{\substack{r \in R_t: \\ tc \in TC^r}} o_{t,r,ref_{r,tc}} + \sum_{tc' \in TC(ref_{r,tc}, tc, r)} (rt_{r,ty_t,tc'} x_{t,r} + l_{t,r,tc'}) \\ + \sum_{\substack{tc' \in \cup_{r \in R_t} TC^r: \\ tc \in OTC_{ty_t, r, tc'}}} l_{t,r,tc'} + (ct_{r,ty_t,tc} + rel_{bs_{r,tc}}) x_{t,r} \quad \forall t \in T \quad tc \in \cup_{r \in R_t} TC^r \quad (17)$$

$$eU_{t,tc} - M(1 - y_{t,t',tc'}) \leq sU_{t',tc} \quad \forall t, t' \in T \quad \text{index } t < \text{index } t' \\ tc, tc' \in \cup_{r \in R_t} TC^r \cap \cup_{r \in R_{t'}} TC^r : \\ tc \in TC_{t,t',tc'} \\ i(t, t') \sum_{r \in R_t} e(tc, r) = 0 \wedge i(t', t) \sum_{r \in R_{t'}} e(tc, r) = 0 \quad (18)$$

$$eU_{t',tc} - My_{t,t',tc'} \leq sU_{t,tc} \quad \forall t, t' \in T \quad \text{index } t < \text{index } t' \\ tc, tc' \in \cup_{r \in R_t} TC^r \cap \cup_{r \in R_{t'}} TC^r : \\ tc \in TC_{t,t',tc'} \\ i(t, t') \sum_{r \in R_t} e(tc, r) = 0 \wedge i(t', t) \sum_{r \in R_{t'}} e(tc, r) = 0 \quad (19)$$

Following the classification used in Section 4.1, constraints (5)–(11) concern the traveling of the trains in the network, constraints (12)–(14) the change of rolling stock configuration and constraints (15)–(19) the capacity of the infrastructure. In more detail, these constraints impose the following conditions.

A train t cannot be operated earlier than $init_t$ (5). The start time of track-circuit occupation along a routing is zero if the routing itself is not used (6). A train starts occupying track-circuit tc along a routing after spending in the preceding track-circuit its longer stay and its travel time, if the routing is used (7). A train t with a scheduled stop at station s and using routing r does not enter the track-circuit following tc before the scheduled departure time from s if tc is in $TCS_{t,s}$ (8), and t has a longer stay in tc of at least $dw_{t,s}$ (9). A train t must use exactly one routing (10). 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 (11).

If trains t' and t use the same rolling stock and t results from t' , a time at least equal to $ms_{t',t}$ must separate their respective arrival and departure times (12). Moreover, the track-circuit tc where the turnaround, join or split takes place (13) 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 (14). 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 arriving train, its arriving being strictly smaller than the one of the second arriving train.

A train's utilization of a track-circuit tc starts as soon as the train occupies the track-circuit $ref_{r,tc}$ along one of the routing alternatives including it, minus the formation time (15). These constraints are imposed as inequalities (16) when they concern a track-circuit of the first block-sections of the routing ($ref_{r,tc} = s_{r,tc_0}$) 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' , constraint (14) ensures 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 $ms_{t',t}$ 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 travel time (see constraint (7)). 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 routing, plus the release time (17). The utilization time includes: the travel 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, the utilization time includes the longer time on all track-circuits tc' such that $tc \in OTC_{ty_t, r, tc'}$. As reported in the notation list, if the head of train t is on one of the track-circuits in $OTC_{ty_t, r, tc'}$, then its tail has not yet exited tc : train

t is longer than tc' , or of the sequence of track-circuits between tc and tc' . Hence, if train t suffers a longer stay when its head is on one of the track-circuits in $OTC_{ty_t, r, tc'}$, such a longer stay must be counted in the utilization time of tc .

Finally, the track-circuit utilizations by two trains must not overlap ([18,19]).

References

- Bussieck, M., 1998. Optimal Lines in Public Rail Transport. Technische Universität Braunschweig (Ph.D. thesis).
- Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., Wagenaar, J., 2014. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B* 63 (1), 15–37.
- Cacchiani, V., Toth, P., 2012. Nominal and robust train timetabling problems. *European Journal of Operational Research* 219 (3), 727–737.
- Caimi, G., Chudak, F., Fuchsberger, M., Laumanns, M., Zenklusen, R., 2011. A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling. *Transportation Science* 45 (2), 212–227.
- Caimi, G., Fuchsberger, M., Laumanns, M., Lüthi, M., 2012. A model predictive control approach for discrete-time rescheduling in complex central railway station approach. *Computers and Operations Research* 39 (11), 2578–2593.
- Carey, M., Carville, S., 2003. Scheduling and platforming trains at busy complex stations. *Transportation Research Part A* 37 (3), 195–224.
- Carey, M., Crawford, I., 2007. Scheduling trains on a network of busy complex stations. *Transportation Research Part B* 41 (2), 159–178.
- Corman, F., D'Ariano, A., Hansen, I.A., Pacciarelli, D., 2011. Optimal multi-class rescheduling of railway traffic. *Journal of Rail Transport Planning and Management* 1 (1), 14–24.
- Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2010. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B* 44 (1), 175–192.
- Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2014. Dispatching and coordination in multi-area railway traffic management. *Computers and Operations Research* 44 (1), 146–160.
- Corman, F., Goverde, R.M.P., D'Ariano, A., 2009. Rescheduling Dense Train Traffic Over Complex Station Interlocking Areas. *Lecture Notes in Computer Science*, vol. 5868. Springer Berlin Heidelberg, pp. 369–386.
- Corman, F., Meng, L., 2015. A review of online dynamic models and algorithms for railway traffic management. *IEEE Transaction on Intelligent Transportation Systems* 16 (3), 1274–1284.
- D'Ariano, A., Corman, F., Pacciarelli, D., Pranzo, M., 2008. Reordering and local rerouting strategies to manage train traffic in real time. *Transportation Science* 42 (4), 405–419.
- 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.
- D'Ariano, A., Samà, M., D'Ariano, P., Pacciarelli, D., 2014. Evaluating the applicability of advanced techniques for practical real-time train scheduling. *Transportation Research Procedia* 3, 279–288.
- Dessouky, M., Lu, Q., Zhao, J., Leachman, R., 2006. An exact solution procedure to determine the optimal dispatching times for complex rail networks. *IEEE Transactions* 38 (2), 141–152.
- Dorigo, M., Stützle, T., 2004. *Ant Colony Optimization*. MIT Press, Massachusetts Institute of Technology, Cambridge.
- Fan, B., Roberts, C., Weston, P., 2012. A comparison of algorithms for minimising delay costs in disturbed railway traffic scenarios. *Journal of Rail Transport Planning & Management* 2 (1–2), 23–33.
- Fang, W., Yang, S., Yao, X., 2015. A survey on problem models and solution approaches to rescheduling in railway networks. In: *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, doi:10.1109/TITS.2015.2446985.
- Ginkel, A., Schöbel, A., 2007. To wait or not to wait? The bi-criteria delay management problem in public transportation. *Transportation Science* 41 (4), 527–538.
- Goossens, J.H.M., Van Hoesel, C.P.M., Kroon, L.G., 2005. On solving multi-type railway line planning problems. *European Journal of Operational Research* 168 (2), 403–424.
- Hansen, I.A., Pachl, J., 2014. *Railway Timetabling & Operations*. Eurailpress, Hamburg, Germany.
- Huang, J., 2006. Using ant colony optimization to solve train timetabling problem of mass rapid transit. In: *Proceedings of the Joint Conference on Information Science* 2006.
- Lamorgese, L., Mannino, C., 2015. An exact decomposition approach for the real-time train dispatching problem. *Operations Research* 63 (1), 48–64.
- Liu, S.Q., Kozan, E., 2009. Scheduling trains as a blocking parallel-machine job shop scheduling problem. *Computers and Operations Research* 36 (10), 2840–2852.
- López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M., 2011. The irace Package, Iterated Race for Automatic Algorithm Configuration. Technical report tr/iridia/2011-004. IRIDIA, Université libre de Bruxelles, Belgium.
- Lusby, R.M., Larsen, J., Ehrgott, M., Ryan, D., 2011. Railway track allocation: models and methods. *OR Spectrum* 33 (4), 843–883.
- Lusby, R.M., Larsen, J., Ehrgott, M., Ryan, D.M., 2013. A set packing inspired method for real-time junction train routing. *Computers and Operations Research* 40 (3), 713–724.
- 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.
- Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an n-track network: a model reformulation with network-based cumulative flow variables. *Transportation Research Part B* 67 (1), 208–234.
- Mu, S., Dessouky, M., 2011. Scheduling freight trains traveling on complex networks. *Transportation Research Part B* 45 (7), 1103–1123.
- 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.
- 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.
- SysML Open Source Specification Project, 2015. Sysml.org.
- Rodriguez, J., 2007. A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B* 41 (2), 231–245.
- Scholl, S., 2005. *Customer-Oriented Line Planning*. University of Kaiserslautern (Ph.D. thesis).
- 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., 2006a. An ant colony optimization meta-heuristic for subset selection problems. *System Engineering using Particle Swarm Optimization*. Nova Science publisher, pp. 7–29.
- Solnon, C., Bridge, D., 2006b. A study of ACO capabilities for solving the maximum clique problem. *Journal of Heuristics* 12 (3), 155–180.
- Stützle, T., Hoos, H.H., 2000. MAX-MIN ant system. *Future Generation Computer Systems* 16 (8), 889–914.
- 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.
- Tsuji, Y., Kuroda, M., Kitagawa, Y., Imoto, Y., 2012. Ant colony optimization approach for solving rolling stock planning for passenger trains. In: *Proceedings of 2012 IEEE/SICE International Symposium on System Integration*. SII, December 16–18. Kyushu University, Fukuoka, Japan.
- Zwaneveld, P.J., Kroon, L.G., Van Hoesel, S.P.M., 2001. Routing trains through a railway station based on a node packing model. *European Journal of Operational Research* 128 (1), 14–33.