



HAL
open science

Formulation of train routing selection problem for different real-time traffic management objectives

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

► **To cite this version:**

Bianca Pascariu, Marcella Sama, Paola Pellegrini, Andrea d'Ariano, Joaquin Rodriguez, et al.. Formulation of train routing selection problem for different real-time traffic management objectives. *Journal of Rail Transport Planning & Management*, In press, 10.1016/j.jrtpm.2024.100460 . hal-04617645

HAL Id: hal-04617645

<https://hal.science/hal-04617645>

Submitted on 19 Jun 2024

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

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

Formulation of train routing selection problem for different real-time traffic management objectives

B. Pascariu^{1,2*}, M. Samà², P. Pellegrini¹, A. D'Ariano², J. Rodriguez¹, D. Pacciarelli²

To cite this paper:

Pascariu, B., Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., & Pacciarelli, D. (2024). Formulation of train routing selection problem for different real-time traffic management objectives. *Journal of Rail Transport Planning & Management*, 31, 100460.

Formulation of train routing selection problem for different real-time traffic management objectives

B. Pascariu^{1,2*}, M. Samà², P. Pellegrini¹, A. D’Ariano², J. Rodriguez¹, D. Pacciarelli²

¹Univ Gustave Eiffel, COSYS-ESTAS, Rue Élisée Reclus 20, 59666 Villeneuve d’Ascq, Lille, France

²Roma Tre University, Department of Engineering, Via della Vasca Navale 79, 00146 Rome, Italy

* Corresponding author email: bianca.pascariu@univ-eiffel.fr

Abstract

The train routing selection problem (TRSP) addresses the optimized selection of alternative routes as a preliminary step for real-time railway traffic management problem (rtRTMP). In the TRSP, route selection relies on estimating potential delays resulting from scheduling decisions. The selected routes are then exclusively applied in the rtRTMP. While prior research established the mathematical model and solution algorithms for the TRSP, its practical application in real-time rail traffic management remains limited. The existing TRSP model focuses on a single objective function for the rtRTMP. However, in practice, various stakeholders may prioritize different objectives, leading to diverse objective functions employed in the rtRTMP. This paper extends the TRSP model by considering a range of suitable objectives for the rtRTMP. We formulate the TRSP for each objective function and enhance the cost estimation model to evaluate the correspondence between the TRSP and rtRTMP objective functions. We then assess the overall effectiveness of the TRSP for the rtRTMP through an evaluation that takes into account several configurations of the model and the rtRTMP solution approach used. Our purpose is to enlarge the applicability of the TRSP and enhance the efficiency of the rtRTMP for real-world systems. The paper includes an in-depth computational analysis of two French case studies to investigate the performance of the TRSP across different rtRTMP configurations.

Keywords: Rail Transportation, Train Routing, Real-time railway traffic management, Performance evaluation

1 Introduction

Railway transport systems play a fundamental role in achieving sustainable mobility. European and international initiatives (e.g., Shift2Rail, Europe’s Rail) strongly promote the increase of railway market shares by encouraging railway infrastructure managers to improve service quality in terms of capacity, reliability, and punctuality. The integration of real-time traffic management systems with advanced technology solutions is one of the key actions to achieve these objectives.

In railway operations, trains run according to a timetable designed beforehand to meet expected demand and ensure effective train services. The timetable defines the passing time, the arrival time and the departure time of trains in stations, the traveling route in the network, and the passing orders over common railway track sections [6]. To account for potential delays in real-time operations, the timetable includes buffer times [10, 13]. However, these buffer times are usually limited to avoid over-consuming the capacity of the network. As a result, delays are not always fully absorbed, which can lead to longer travel times, conflicts between trains, and a decrease in the overall efficiency and quality of service of the railway system.

In operations control centers, human dispatchers are in charge of minimizing the conflicts and the impact of delays, typically using retiming, reordering, and rerouting actions. The problem is known as the real-time railway traffic management problem (rtRTMP). Several models and algorithms have been proposed in the literature to solve the rtRTMP and provide support to dispatchers, helping them take more informed decisions [3]. Nevertheless, how to best solve the

rtRTMP is still an open question. The rtRTMP has been proved to be NP-hard [18], with the number of alternative routes available for each train strongly affecting the search space size [16].

Considering alternative routes helps minimize train conflicts and delay impact, ensuring that the railway system operates as efficiently as possible [19, 35]. However, routing variables exponentially increase the number of solutions to explore [16]. This in turn raises the computational cost of finding high quality solutions [7, 24] in advanced decision support systems that rely on mathematical optimization methods.

To reduce the computational burden and preserve the advantages of train rerouting, a strategy that is gaining track is restricting the number of alternative routes considered in the rtRTMP. Samà et al. [28] shows the benefit of systematically studying which subsets of routes to select for each train, introducing in the literature a specific problem, known as the train routing selection problem (TRSP). The TRSP represents a new research area in the railway traffic management field. Samà et al. [28] model the TRSP, propose a procedure to estimate the cost of choosing a route in the rtRTMP based on the impacts of the related scheduling decisions and solve the problem using a customized Ant Colony Optimization (ACO) algorithm [11, 31], which is often able to converge to global optimal solutions [21]. However, the algorithm struggles to find high-quality solutions in real-time when the size of the instances increases. Pascariu et al. [22] overcome this issue by developing a parallel ACO-TRSP algorithm to speed up and diversify the search space exploration in the available computation time. They also improve the cost estimation procedure by including the impact of rolling stock constraints and potential delay propagation.

This recent research on the TRSP sets up the main points for its mathematical model and the algorithms used for its solution process, but there are still some open issues. These studies have primarily focused on minimizing the total delay of trains at their destination as objective function for the rtRTMP, limiting to the applicability and proven effectiveness of the TRSP. Railway traffic management is marked by a multitude of stakeholders, including train operators, network managers, and passengers, each with their distinct priorities and interests. Moreover, railway systems exhibit substantial operational diversity. As a result, there is a wide range of objective functions used for the rtRTMP both in the existing literature [27] and in practice. Thus, it is necessary to evaluate if the TRSP can maintain its effectiveness when the rtRTMP addresses a broad spectrum of objectives. Furthermore, how should the TRSP model be adjusted to the different objectives? A study is therefore needed to harmonize the TRSP with the different priorities and operational realities encountered in railway systems, which translates in different objectives optimized in the rtRTMP.

Furthermore, previous studies have only tested the effectiveness of solving the TRSP to feed one specific rtRTMP solver, RECIFE-MILP [24], which uses a mixed-integer linear programming (MILP) formulation [23] and a truncated exact algorithm. It is important to investigate the effectiveness of the TRSP application when using different rtRTMP solvers and optimization techniques, such as heuristic-based approaches, metaheuristics, or other optimization methods, to determine the potential of the TRSP for practical implementation in real-world railway traffic management systems, regardless of how the rtRTMP is solved.

In this paper, we aim to generalize the use of the TRSP, regardless of which objective function is optimized in the rtRTMP and how the problem is modeled or solved. More in-depth:

- we study how to best achieve a correspondence between the TRSP and rtRTMP from a modeling point of view, given how varied the rtRTMP literature is in terms of objective functions related to delay minimization. These objectives include not only minimizing the total delay of trains at their destination, as already done, but also minimizing the total delay both at the entry and exit points of the network, the total train travel time, the number of delayed trains, the maximum delay at the destination, and the maximum completion time;
- we aim to broaden the applicability of the TRSP and evaluate its general effectiveness on the rtRTMP regardless of how alternative routes are accounted for in the rtRTMP and how the problem itself is solved. By TRSP effectiveness we mean the ability to select routes that help an rtRTMP solver finding better quality solutions more quickly than when used on its own. Indeed, in the literature, some approaches consider routing variables simultaneously with

rescheduling ones [19, 34], while others define a sequential order between them, decomposing the overall problem [7, 36].

Considering the first point, we propose TRSP formulations that reflect the different rtRTMP objective functions taken into account in the literature and extend the cost estimation model of Pascariu et al. [22] accordingly. This allows us to evaluate how to best achieve a correspondence between the TRSP and rtRTMP objective functions in order to identify the routes that lead to the best rtRTMP solutions. We solve the TRSP model, as formulated, through the parallel pACO-TRSP algorithm introduced by Pascariu et al. [22]. It is important to note that each TRSP objective function corresponds to a distinct problem formulation. To address these differences effectively, we customize part of the algorithm to dynamically adjust its search strategy according to the specific objective function in use. Regarding the second point, besides the model from Pellegrini et al. [23] solved with RECIFE-MILP [24], previously used in the considered stream of research, we look at the decision support tools AGLIBRARY. AGLIBRARY models the problem as an alternative graph [18], and solves the rerouting problem iteratively, using a meta-heuristic framework where, at each step, a new promising route assignment is selected to improve the current solution. We chose AGLIBRARY, the optimization core of ROMA [7, 9, 30], and RECIFE-MILP, since they demonstrated their high potential in the European project ON-TIME [26]. The purpose is not to compare the two rtRTMP approaches, but each method is used within the proper setting (objective function) for which it is most appropriate. In particular, RECIFE-MILP is used with cumulative objective functions, while AGLIBRARY is used with minimization of maximum values.

We conduct a comprehensive set of experiments on realistic instances of two French test cases: the railway line around the city of Rouen and the station area of Lille Flandres. Overall, our experiments will provide a comprehensive evaluation of the applicability and robustness of the TRSP model in real-time railway traffic management.

The rest of the paper is structured as follows. Section 2 reviews the relevant literature on TRSP. Section 3 provides a general description of the problem, focusing on the objective functions taken into account for the rtRTMP and on the related problem characteristics that need to be reflected in the TRSP. Section 4 presents the formulation of the TRSP search space, while Section 5 reports the required extensions to take into account the different objective functions proposed. Section 6 describes the solution approach from Pascariu et al. [21] used for the TRSP, with the customization needed for the different objective functions. Section 7 shows the computational results and their analysis, and Section 8 summarizes the conclusions and suggests where to focus future research.

2 Literature review

The rtRTMP is a topic that has received significant research attention [3] since it belongs to the class of NP-hard problems [18]. Its combinatorial complexity arises from the large number of possible train routes and the interactions between trains. Specifically, as the number of available routes grows, the number of scheduling combinations increases exponentially, enlarging the problem size and its computational complexity.

The TRSP directly addresses this computational complexity: it selects a limited set of alternative routes for each train, which are then used in the rtRTMP as the only possible available alternatives. The TRSP was only recently formalized by Samà et al. [28] by a construction graph, together with an integer linear program and a model to estimate the costs due to route choices in the rtRTMP, in terms of conflict occurrence and scheduling decision impacts. Moreover, they developed an ACO algorithm [11, 31] to solve the problem. This algorithm has been shown to converge to globally optimal solutions in Pascariu et al. [21]. However, it struggles to find high-quality solutions in real-time when the size of the instances increases. Recently, the research on TRSP [22] focused on improving the model and algorithm of Samà et al. [28] for large instances with rolling stock re-utilization constraints. In these first studies [22, 28], the TRSP was modeled to select routes minimizing the objective function of a state-of-the-art MILP-based solution approach [24], i.e., the total train delay at the exit from the infrastructure.

The rigorous study of the impact and potential criteria related to limiting routing numbers, as

conducted in the TRSP, is not frequent in the existing literature. Whereas, it is worth noting that the rtRTMP is addressed by different methodologies in the literature. This diversity is reflected in the plurality of models and the absence of a universally accepted objective function for the problem [27]. To provide a comprehensive overview, Table 1 compares recent studies on the rtRTMP based on key aspects such as type of model used, recovery measures considered, alternative route processing, objective function, and solution algorithms. We observe from the table that only the two mentioned studies [22, 28] apply optimized processing of alternative routes via the TRSP. Other methods either address the computational complexity within the rtRTMP solution process itself or limit the number of alternative routes available in the rtRTMP in a non-optimized form.

A possibility to handle rerouting variables in the rtRTMP is to consider them simultaneously with retiming and reordering (rescheduling) variables and use an enhanced solution algorithm that can cope with the large number of variables. Pertaining to this research branch, Pellegrini et al. [23] uses a MILP model to minimize train exit delays, with a heuristic algorithm, specifically a real-time boosted truncated exact algorithm [24]. The algorithm first addresses the train rescheduling problem using timetable routes and then refines it by exploring retiming, reordering, and rerouting variables. Meng & Zhou [19] proposes a cumulative flow variable-based model for the simultaneous train rerouting and rescheduling problem. The model is formulated as an integer program (IP) minimizing the total completion time of trains and is solved via a Lagrangian relaxation solution framework. Toletti et al. [34] use a resource conflict graph-based decomposition [4] and a coordination framework to minimize the total train delays. In particular, the network is split in sub-networks represented microscopically, and traffic in all sub-networks is coordinated considering a macroscopic infrastructure representation. The authors utilize a MILP commercial solver for the macroscopic level and an ad-hoc column generation method for the microscopic one including train routing variables. Lu et al. [17] use a MILP formulation to minimize the total train delay and platform assignments. Yuan et al. [38] use a MIP formulation to improve the punctuality and regularity in train operations, reduce the passenger waiting time, and alleviate the passenger flow burden of platforms, where the dynamic coupling interactions among network-wide passengers, trains, and stations are systematically considered. Zhang et al. [37] define a ILP model, based on a space–time network, to minimize the number of abandoned passengers and the total passenger delay when a large disruption occurs. They include the option of rerouting only the trains directly affected by the disruption and opting for alternative routes to avoid it. To reduce the computational complexity, both Lu et al. [17], Zhang et al. [37] and Yuan et al. [38] reduce the number of variables by decomposing the problem into smaller time horizons in which rerouting measures are simultaneously tackled with the rescheduling ones, including fewer trains.

Other approaches handle rerouting variables by sequentially determining train routes and schedules. We highlight in Table 1 the order in which the decisions are taken using the logic symbol \rightarrow . Corman et al. [7] and Samà et al. [29] use the alternative graph [18, 9] to minimize the maximum exit train delay, and iteratively solve the train rescheduling and rerouting problems separately. Given the rescheduling solution obtained by a truncated branch-and-bound [9], only specific trains are selected to be rerouted by a tabu search [7] or a variable neighborhood [29] schemes, picking the routes randomly one at a time. Based on a hybrid model mixing disjunctive and alternative graphs, Gholami & Törnquist [14] minimize the total exit train delay via a heuristic algorithm that first applies rerouting and then rescheduling. Similarly, in Van Thielen et al. [36], whenever a conflict is located in a station area, first rerouting is applied based on a flexible job shop scheduling problem. If this rerouting fails to find a zero-delay solution or if conflicts occur outside station areas, they pursue rescheduling, focusing on variables associated with trains affected by the original timetable perturbation not attributable to delay propagation. Caimi et al. [5] propose a model predictive control-based solution approach for dispatching trains in complex railway stations by assigning predetermined blocking stairways to trains as in Caimi et al. [4]. They limit the number of routes in an offline step to the rtRTMP solution, selecting the fastest ones based on the experience of infrastructure experts. Bettinelli et al. [2] propose an algorithm based on the repeated execution of a greedy approach that schedules trains on a time-spaced network, considering a limited set of predefined alternative routes for each train. In the objective function, they minimize the total penalty given by the cost of delays or longer travel time routes. All these rtRTMP solution ap-

Table 1: Summary of rtRTMP solution approaches. Models: mixed-integer linear programming (MILP), mixed-integer programming (MIP), integer linear programming (ILP), integer programming (IP), alternative graph (AG). Recovery measures: retiming (T), reordering (O) and rerouting (R), simultaneous processing (+), sequential processing (\rightarrow).

Paper	Model	Recovery measures	Alt. routes processing	Objective function (minimized)	Solution Algorithm
Pellegrini et al. [23, 24]	MILP	T+O+R	No	Total exit train delay	Commercial solver, heuristic [24]
Samà et al. [28], Pascariu et al. [22]	MILP [23]	T+O+R	TRSP	Total exit train delay	Commercial solver, heuristic [24]
Meng & Zhou [19]	IP	T+O+R	No	Total train completion time	Lagrangian heuristic, commercial solver
Toletti et al. [34]	ILP	T+O+R	No	Total train delay	Commercial solver, column generation
Lu et al. [17]	MILP	T+O+R	No	Total train delay and platform assignment	Commercial solver, heuristic
Fischetti & Monaci [12]	MILP	T+O+R	Random subset	Average train delay	Commercial solver, heuristic
Zhang et al. [37]	Space-time network, ILP	R+T+O	No	Total passenger delay and abandoned passengers	Lagrangian heuristic
Corman et al. [7]	AG	T+O \rightarrow R	No	Maximum exit train delay	Branch and bound, tabu search
Samà et al. [29]	AG	T+O \rightarrow R	No	Maximum exit train delay	Branch and bound, variable neighborhood
Gholami & Törnquist [14]	AG	R \rightarrow T+O	No	Total exit train delay	Heuristic
Van Thielen et al. [36]	ILP	R \rightarrow T+O	No	Total train delay	Heuristic
Caimi et al. [5]	ILP	R \rightarrow T+O	Experience-based subset	Passenger dissatisfaction caused by delays	Commercial solver
Bettinelli et al. [2]	Time-space graph, MIP	R \rightarrow T+O	Predefined set	Total penalty (train travel time and delay)	Heuristic

proaches [2, 5, 20] are based on the assumption that a partial routing flexibility is to be preferred to a complete routing flexibility in the rtRTMP.

Overall, the mentioned literature shows that routing variables are critical, whatever solution approach is used. When retiming, reordering, and rerouting are tackled simultaneously (T+O+R), the number of routing variables strongly affects the problem size and the required computation time. Instead, when rerouting and rescheduling variables are sequentially optimized (R \rightarrow T+O or

T+O→R), the routes are selected randomly between all the alternative ones. In principle, the TRSP can be applied in combination with both the simultaneous and sequential ways of handling routing variables. In the first one, limiting the number of routing variables can help to improve the exploration of scheduling solutions. In the second one, an accurate pre-selection and initial evaluation of routes can orient the search in solving the rtRTMP. However, as mentioned at the beginning of this section, a systemic study on the improvement that the TRSP may bring to the runtime or objective value of the rtRTMP solution, in general, has yet to be done: so far, the state-of-the-art TRSP preprocessing has been applied only to a single rtRTMP solver, in which rerouting and rescheduling decisions are taken simultaneously to minimize total delay. Its effectiveness when applied in combination with different methods for solving the rtRTMP is unknown. Furthermore, the state-of-the-art TRSP model has been developed for a specific objective function of the rtRTMP, while it is worth noting that the rtRTMP is addressed by a plurality of models and objective functions [27], as also demonstrated in the analysis of the literature provided in this section. The general usefulness of the TRSP in this sense is still unexplored.

3 Problem description

Typically the railway network is divided in control areas, consisting of the railway infrastructure, the signaling and the safety systems within a limited geographical area. Under the fixed block signaling system, the infrastructure is divided by electrical *track circuits*, which identify the minimum sections necessary to detect the presence of a train. The set of track circuits between two consecutive signals is referred to as *block-section*, which typically defines the space a train needs to brake and stop. This holds in particular in three-aspect signaling systems, which are the most commonly deployed.

The operations of each train in a control area follow a timetable, which specifies: the route to be used by the train, the locations where the train must stop (*stopping points*), and the arrival, departure, and dwell times at the stopping points. The *train route* defines the set of track circuits the train traverses from its *entry point* in the control area to its *exit point* (or destination). These points correspond, respectively, to the first and the last track circuit the train traverses. The *dwell time* is the time interval that each train is scheduled to stop at a stopping point. The *train travel time*, relative to a specific route, corresponds to the time span starting from the physical occupation of its first track circuit up to the end of its last track circuit occupation. The *infrastructure utilization* is the time during which a track circuit is assigned solely to a particular train and is inaccessible to other trains. The utilization of a track circuit by a specific train is reserved before its physical occupation to ensure that the track circuit cannot be entered by other trains for the time the current train is scheduled to traverse it. We consider a *route lock-sectional release* interlocking system, which involves the simultaneous reservation of all track circuits within a block-section. These track circuits are subsequently released one by one, based on train transit and track-free detection.

The signaling system ensures safe train operations by establishing a minimum duration between two trains to prevent any overlap in infrastructure utilization, i.e., the *minimum headway time* [15]. To guarantee the train minimum headway times and to prevent unexpected brakings, the three-aspect signaling system provides the simultaneous reservation of the two block-sections following a free-way signal. Thus, the utilization time of a track circuit is usually larger than its physical occupation, because it includes *reservation* and *release times*.

3.1 The real-time railway traffic management problem

In real-time operations, the scheduled train infrastructure utilization is susceptible to perturbations due to unexpected events such as mechanical malfunctions, infrastructure issues, adverse weather, or passenger boarding times that are longer than expected. These perturbations (that generate *primary delays*) can generate conflicts with other trains in the control area. A *conflict* occurs when two trains traveling at their scheduled speed are expected to use the same track circuits for an overlapping time period.

The rtRTMP formalizes the problem faced by dispatchers of detecting potential train conflicts and minimizing their impact by means of train retiming, reordering, and rerouting decisions. Thus, the rtRTMP solution aims to provide a recovery plan that avoids any deadlocks and ensures that each track circuit is utilized by only one train at any given time, meeting the minimum headway time required by the safety systems. In the rtRTMP, ordering decisions consist of defining train passing orders for each common track section, while routing decisions consist of selecting train routes traversing the network among all the *train alternative routes*. The alternative routes share the same stopping points and the same initial and final track circuit with the timetable route while passing through different sequences of track circuits. Available alternative routes may also depend on further compatibility issues (e.g., platforms long enough for a specific rolling stock or with specific configurations for specific passenger flows) and the infrastructure manager defines them. The timing decisions allowed during the rtRTMP solution involve: stopping a train at a signal along the line or postponing the time (of arrival/departure) at the stopping points or at the entry or exit points from the control area. If these actions lead a train to arrive at the relevant points later than the planned arrival time, then the train is considered late. In this case, the train delay is called *secondary delay* since it is a consequence of the actions taken to solve the conflict caused by the initial perturbation. Furthermore, we distinguish between *exit delay* and *entry delay*. The former is the positive deviation compared to the planned exit time, while the latter is the positive deviation from its scheduled entry time. Secondary delays easily propagate across the network between trains, thus optimization methods are applied to solve the rtRTMP to minimize the effects of initial timetable perturbations, in terms of various objective functions.

3.1.1 Objective functions

In practice, the choice of objective functions and their relative importance may vary depending on the specific railway system, regulatory requirements, and operator preferences. In this paper, we focus on the objective functions oriented to train operations. We describe below some of the most widely used in the related literature.

- *Total Exit Delay - TED*: the minimization of the sum of all train delays at their exit points from the control area. This objective is relevant because it minimizes the delay at the train destination or the next control area will have to deal with it;
- *Total Delay - TD*: the minimization of the sum of all train delays, both at the entry and exit points from the control area. Compared to TED minimization, this objective function includes entry delays because, by delaying the moment in which a train is taken in charge, one may add a burden to the previous control area traversed by the train itself. Hence, in ideal traffic management, both entry and exit delays should be minimized;
- *Total Travel Time - TTT*: the minimization of the aggregated travel time of all trains in the control area. In railway practice, this objective function is useful mostly when dealing with freight traffic, for which it may be preferable to depart and arrive later than breaking, stopping, and re-accelerating along the route. Indeed, these actions would require great energy and time consumption;
- *Number of Delayed Trains - NDT*: the minimization of the total number of trains that exit late the control area. This objective aims to limit the number of trains experiencing delays, which can help maintain a smoother and more predictable service for passengers and operators;
- *Maximum Delay - MD*: the minimization of the maximum train delay at the exit point from the control area. This objective function aims to limit delay propagation and ensure fairness among trains, by preventing a single train from experiencing excessively long delays, even if it means some trains have slightly increased ones;
- *Maximum Completion - MC*: the minimization of the time at which the last train ends its journey. This objective function aims to optimize the use of the infrastructure to accommodate as soon as possible future trains.

3.2 Train Routing Selection Problem

The TRSP addresses the problem of choosing the alternative train routes that help an rtRTMP solver in finding better quality solutions more quickly than when used on its own: the TRSP represents a preprocessing and has to be solved before starting the solution of the rtRTMP, where only the routes selected will actually be considered. Since the rtRTMP is solved in real-time and the efficacy of route choice depends on the given traffic situation, the TRSP has been shown to be of maximum benefit when considered as a real-time problem [30].

A solution for the TRSP has to be feasible for the rtRTMP. Indeed, from a routing point of view, infeasibility in the rtRTMP may stem from assigning routes to trains that do not satisfy rolling stock re-utilization operations. The feasibility of a train route combination is verified when the chosen train routes are *coherent* with the rolling stock turnaround, joining, or splitting operations. Specifically, the routes of trains involved in rolling stock re-utilization operations are coherent if they occur on the same infrastructure section. To provide a proper counterbalance between computational efficiency and high quality solutions, the TRSP considers a simplified model to assess route impact on the rtRTMP. Train timing and ordering are assessed in the TRSP only to estimate the effect of the selected train routes. Timing and ordering variables are not explicitly considered in the TRSP, but still need to be evaluated to understand the potential effect of a specific route choice. This is obtained through the use of cost estimations on the potential delays that would occur if the selected routes were used in the rtRTMP solution. These costs associated with train routes are determined based on their impacts on schedules and how they interact, aiming to make the best local scheduling decisions. The cumulative effects of the different route choices need then to be evaluated based on the actual objective function optimized during the rtRTMP solution process. To best achieve this, specific aspects related to different functions may be included in the TRSP model. In Section 4, we present the general TRSP model and, in Section 5, how we extend it in order to improve its correspondence to the objective function considered in the rtRTMP.

4 TRSP model

We model the TRSP through a k -partite construction graph $G = (V, E)$ [28]. A vertex $v \in V$ represents an alternative train route. The vertices are grouped into k partitions, with k the total number of trains. For each train $t \in T$ requiring to traverse the considered railway infrastructure in a certain time window all its alternative route assignments belong to a partition, i.e., the independent set $V_t \subset V$, with $\cup_{t \in T} V_t = V$ and $\cap_{t \in T} V_t = \emptyset$. Two vertices $v_i, v_j \in V$ are connected by an edge $e_{ij} \in E$ if they represent coherent routes and are assigned to different trains. A feasible combination of train routes for the rtRTMP is represented by a k -vertex clique c in the TRSP. Let us remind that a clique in G is an induced complete subgraph of G . The clique c can be represented using the following ILP:

$$c = \left\{ \begin{array}{l} x \in \{0, 1\}^{|V|} \\ y \in \{0, 1\}^{|E|} \end{array} : \begin{array}{ll} \sum_{v_i \in V_t} x_i = 1 & \forall V_t \subset V \quad (1.a) \\ \sum_{v_i \in V_t} y_{ji} = (t-1)x_i & \forall v_j \in V_s : s = \{1, \dots, t-1\} \quad (1.b) \\ \sum_{v_i \in V_t} y_{ij} = (k-t)x_i & \forall v_j \in V_s : s = \{t+1, \dots, k\} \quad (1.c) \end{array} \right\} \quad (1)$$

We use the binary decision variables x_i and y_{ij} to select, respectively, vertices $v_i \in V$ and edges $e_{ij} \in E$: $x_i = 1$ if $v_i \in c$ (0 otherwise) and $y_{ij} = 1$ if $e_{ij} \in c$ (0 otherwise). The edges in G are not-oriented, i.e., $e_{ij} = e_{ji}$ both corresponding to the edge connecting v_i and v_j . Constraints (1.a) ensure that exactly one route is assigned to each train. Constraints (1.b) and (1.c) ensure that only the edges connected to the selected nodes are chosen. Constraints (1.b) and (1.c) refer to the edges connecting vertex v_i in the partitions $V_t \subset V$, respectively, with those in the preceding and following partitions. Here, the subsets V_t are sorted based on the order in which trains enter the infrastructure.

Consider the set of all cliques on G consisting of k vertices, denoted by Γ . Vertex and edge costs are used in the TRSP to select the subset $S_p \subset \Gamma$ of the p minimum cost cliques following the objective function in Eq. (2), where $f(c)$ is the clique cost.

$$\min \sum_{c \in S_p} f(c) \quad (2)$$

The clique cost needs to be linked to how solutions are evaluated in the rtRTMP. In Section 5, we elaborate on how the correspondence between the TRSP and the rtRTMP objective function can be achieved. This cost, of course, needs also to reflect the quality of the solution it relates to. To this purpose, we associate to each vertex v_i and edge e_{ij} in G a cost, respectively, $u_i : V \rightarrow \mathbb{N}$ and $w_{ij} : E \rightarrow \mathbb{N}$. These costs are determined according to the model proposed by Pascariu et al. [22]. The cost u_i assigned to each vertex v_i in G accounts for the potential train delay due to the increased travel time required for the selected route $v_i \in V_t$ compared to the default $v_d \in V_t$, typically the one assigned in the timetable. This potential delay corresponds to the difference between the train unperturbed running time along v_i and the one along v_d , when it is non-negative.

The edge cost w_{ij} , instead, represents the additional running time and thus the potential delay due to the train scheduling decision when v_i and v_j , linked by e_{ij} , are simultaneously considered. Scheduling decisions are required when different trains have common route sections on which they can potentially have a conflict. Each common section is formed by a set of track circuits. The cost w_{ij} includes two components: (i) a fixed component, assigned to each edge of G , which represents the potential delay caused by the overlap of two train routes at a time (i.e., when the corresponding trains are considered as the only ones in the control area); (ii) a clique-dependent component, added to the fixed component after each clique has been built, which estimates the potential delay propagation due to the interaction of all train routes in the clique.

The overall cost w_{ij} for each pair of train routes v_i, v_j in G , linked by e_{ij} , is computed as follows:

1. The set of all common sections between the considered pair of routes is examined. If this set is empty, no conflict can arise between the concerned train routes, and w_{ij} is set to zero. Otherwise (if the set of common sections is not empty), for each train t in the pair, the utilization time overlap is determined on each common section in case t passes second. This time corresponds to the difference between the end of the utilization of the first train and the start of the utilization of the second one. A positive utilization overlap is the time that t should wait before entering each common section in order to avoid a conflict on it. A negative value instead indicates no overlap: if running on a free network, the two trains traverse the common section at different times;
2. For each train in the given pair, the maximum utilization overlap is selected among all the common sections; this corresponds to the most penalizing section for that train. Then, to determine the fixed component of w_{ij} for the considered pair of train routes, we select the smaller value from the previously calculated maximum utilization overlaps on these two routes. This decision represents the optimized scheduling decision minimizing the potential delay for the trains on these routes. We refer to the second-passing train on the most penalizing selected section as the *potential waiting train*. The fixed component is used as edge cost when vertices are selected to form a clique. The costs w_{ij} of the train routes with common resources but no utilization overlap (fixed component with negative value) is set to one. This is meant to reward pairs of routes where no common sections exist during the clique construction process;
3. After a clique c has been selected, the negative value of the fixed component is used for the clique-dependent cost computation. During this computation, each cost $w_{ij} > 0$ assigned to $e_{ij} \in c$, connecting $v_i, v_j \in c$, is propagated as follows. Let us consider v_i as the route related to the potential waiting train, w_{ij} is propagated to the edges connecting v_i with any other vertex $v_h \in c$ which has common sections with it. This generates the additional clique-dependent cost component on edge $e_{ih} \in I(v_i)$, where $I(v_i)$ is the set of incident edges in v_i . When edge $e_{ih} \in c$ has a fixed component $w_{ih} > 0$, the clique-dependent component corresponds to the non-negative difference between the propagated and the current potential delay, which is added to w_{ih} , so $w_{ih} = \max(w_{ih}, w_{ij})$. Otherwise, when $w_{ih} < 0$, the

propagated cost w_{ij} is added to w_{ih} . That is, the estimated time difference between the two involved trains is reduced, possibly causing a utilization overlap on the most penalizing common section chosen. In the potential delay propagation, the edges are evaluated in the sequence corresponding to the order of train entry in the control area. Once the delay propagation is completed, we still set to one the cost w_{ij} of the edges connecting two train routes that share common sections but do not have any potential delay.

Example

We present a small example to illustrate how to build the TRSP model and compute the costs associated with its vertices and edges. Let us consider four trains (A, B, C and D) traveling in a network, each with a given set of available alternative routes, as presented in Figure 1a. Trains A and B traverse the network from left to right, while C and D travel in the opposite direction. The routes planned in the timetable are $\{a_3, b_1, c_3, d_2\}$, and are considered as default. Moreover, we assume that the rolling stock of train B is reutilized for train D, performing a turnaround in the terminal station of B. Thus, the routes used by B and D must be such that the former terminates where the latter starts. The correspondent construction graph is shown in Figure 1b. A partition in the graph is the set of all feasible alternative train routes, where each vertex is represented by a dot and corresponds to one of the alternative routes. The graph is incomplete. In the case of rolling stock re-utilization, only the vertices corresponding to coherent train routes can be connected. Recall that coherency implies that the train routes share the same track section where the re-utilization operation is carried out. In our example, this involves the last track section of B and the first track section of D. In particular, the last track section of b_1 and b_3 differs from the first track section of d_3 and d_4 . Consequently, train routes b_1 and b_3 cannot be used in combination with d_3 and d_4 and thus are not connected to them. The same applies to the combination of train routes b_2 and b_4 with d_1 and d_2 .

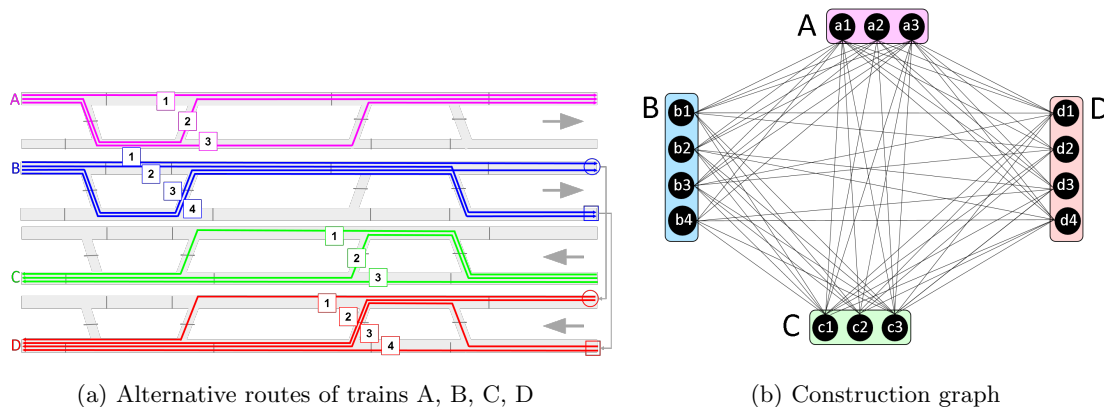


Figure 1: Illustrative example TRSP model

Once the graph is constructed, we calculate the vertex cost and the fixed component of the edge cost for all vertices and edges in the graph. For the sake of brevity, we select four train routes and describe the cost computation for them only. In particular, we consider the train routes $\{a_1, b_2, c_2, d_3\}$ in Figure 2a forming a four-vertex clique c : each train route belongs to a different partition and is connected to the others. Furthermore, for the sake of clarity we assume the following simplifications. The infrastructure is divided into ten sections, labeled from s_1 to s_{10} , each corresponding to a single block-section and usable in both directions. All reservation and release times are null and a two-aspect signaling system is deployed. Thus, the utilization of each section starts when a train enters it and ends immediately when the train leaves it. Thus, when a train finishes occupying one section, the following train can immediately enter it.

We assume a perturbed traffic situation in which train A has an entry delay of 8 time units and train C has an entry delay of 2 time units. Because of these primary delays, potential conflicts

appear. Figure 2b shows the utilization time of all trains on each section when considering each train traveling independently from the others, on a free network. Each line corresponds to one section, and colored rectangles show when each of them is utilized by each train. The potential conflicts are indicated in the picture as rectangles filled with diagonal lines of the colors corresponding to the involved trains. Remark that the turnaround between train B and D requires 3 time units as processing time on s_{10} (indicated by the white rectangle with red dots). Also, due to this turnaround, between the end of utilization of B and the beginning of utilization of D, s_{10} cannot be occupied by another train.

We next illustrate the computation of the cost of vertices. Assuming the minimum travel time from the utilization diagrams in Figure 2b, Table 2 shows, for each train, the minimum travel time r required to traverse the chosen routes, the minimum travel time r_d required for the default (timetable) routes, and the resulting vertex cost u . The minimum travel time is the time it would take the train to traverse the route, without conflicts, defined as the difference between the end of utilization of the last section and the beginning of utilization of the first section. Train routes a_1 and d_3 are shorter than the corresponding timetable route, respectively, by 2 ($|21 - 23|$) and 4 ($|16 - 20|$) time units, thus leading to no potential delay $u_{a_1} = u_{d_3} = 0$. On the contrary, b_2 and c_2 are longer than the corresponding timetable route, respectively, by 1 ($20 - 19$) and 3 ($24 - 21$) time units, thus choosing these alternative routes leads to the potential delays $u_{b_2} = 1$ and $u_{c_2} = 3$.

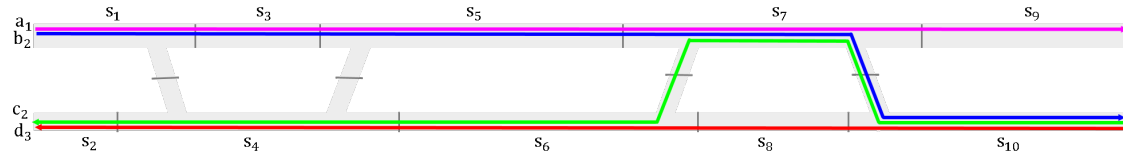
Table 2: Train travel time and vertex costs

	A	B	C	D
r	21	20	24	16
r_d	23	19	21	20
u	0	1	3	0

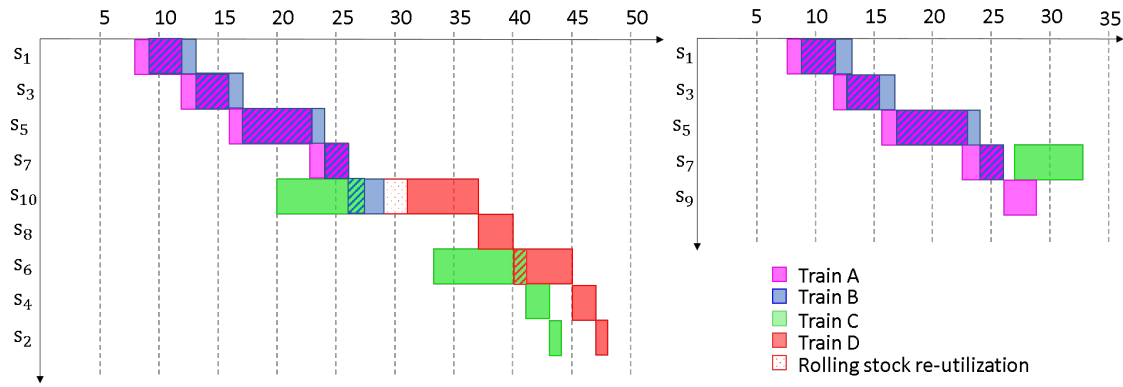
We next determine the costs w associated with each edge connecting the selected train routes. This cost represents the potential delay due to the scheduling decisions taken to solve the conflicts in Figure 2b. As detailed in Section 4, the edge cost consists of two components. The first is a fixed component, which is calculated for pairs of train routes. The second component, known as the clique-dependent component, is added to the fixed component after each clique has been built. This component accounts for the potential delay propagation resulting from the interaction of all train routes within the clique. Hereafter we refer to the fixed component as w^f and to the final edge cost within the clique as w .

Let us consider the edge a_1b_2 , to illustrate the computation of the fixed component. As shown in Figure 2b, the train routes connected by these edges share sections s_1 , s_3 , s_5 and s_7 . If train A passes before B on each of these sections, the time that train B potentially has to wait is given by the utilization overlap calculated as the difference between the end of the utilization of each section by train A and the beginning of its utilization by B. The value obtained for each shared section is, respectively: 3 ($12 - 9$), 3 ($16 - 13$), 6 ($23 - 17$), and 2 ($26 - 24$). Instead, when B passes first, the time that train A potentially has to wait is, respectively: 5 ($13 - 8$), 5 ($17 - 12$), 8 ($24 - 16$), and 3 ($26 - 23$). Thus, for each of the two possible train orderings, the maximum utilization overlap is 6 (A before B) and 8 (B before A). In Figure 2c we illustrate the utilization diagram of the two potential scheduling decisions. The minimum between the maximum utilization overlap in the two configurations gives the fixed component of the potential delay $w_{a_1b_2}^f = 6$. In this case, train B is the potential waiting train.

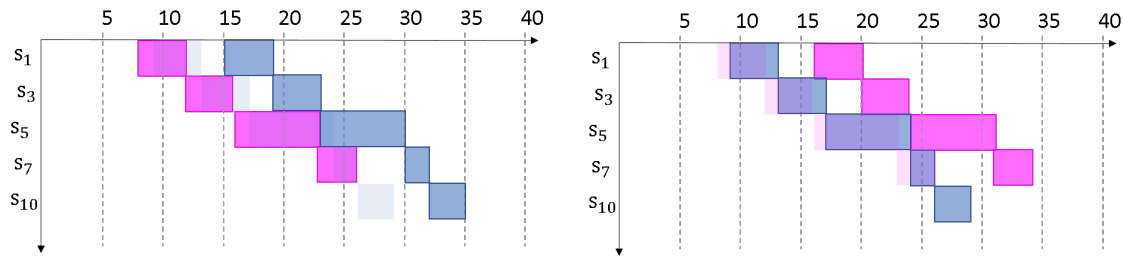
Looking at the train routes a_1 and c_2 , they only share section s_7 , which is traversed in opposite directions by the two trains. Figure 2d (left) shows that when train A is first on this section, the utilization overlap is -1 since train A ends utilizing it at 26 and C starts the utilization at 27. The negative value indicates that C would not be potentially delayed, but rather that some time elapses between the end of the utilization of A and the beginning of C. On the contrary, when train C is first (Figure 2d right), the utilization overlap is 9 ($33 - 24$). Since $\min(-1, 9) = -1$ but the pair of routes has a common section, we still set the fixed component of the potential delay $w_{a_1c_2}^f = 1$, with train C being the potential second one. Train routes a_1 and d_3 have no common section, thus



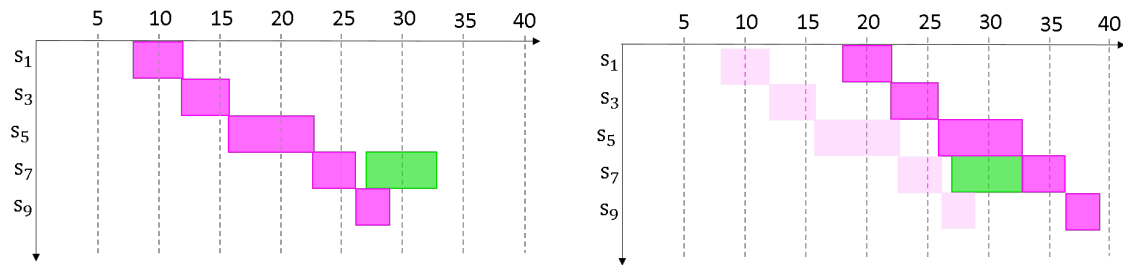
(a) Selected altrain routes



(b) Trains utilization diagram



(c) Trains ordering A-B (left) B-A (right)



(d) Trains ordering A-C (left) C-A (right)

Figure 2: Illustrative example

$w_{a_1d_3}^f = 0$. For the other edges linking the considered train routes, we report in Appendix A the detailed computation of these costs. However, for the sake of illustrating the computation of the clique-dependent component, we report here the results obtained for these edges (in brackets the second passing train): $w_{b_2c_2}^f = 9$ (B), $w_{b_2d_3}^f = 0$ (D), $w_{c_2d_3}^f = 1$ (D).

Let us consider the selected routes form a clique. For the computation of the clique-dependent component, we then assume a knock-on propagation of potential delays, according to the order in which trains enter the infrastructure. Thus, we consider the first potential delay to be propagated being $w_{a_1b_2} = 6$, suffered by train B. It impacts the edges connecting b_2 with c_2 and d_3 . The edge cost $w_{b_2c_2} = 9$ remains unchanged since $\max(9, 6) = 9$, while $w_{b_2d_3} = \max(0, 6) = 6$. In the same way, $w_{b_2c_2} = 9$ suffered by train B is propagated to the edge connecting b_2 with d_3 , resulting in $w_{b_2d_3} = 9$. Table 3 presents the final potential delays obtained for each edge of the considered clique, indicating in brackets the potential waiting train.

Table 3: Edge costs, in brackets the potential waiting train

	a_1b_2	a_1c_2	a_1d_3	b_2c_2	b_2d_3	c_2d_3
w^f	6 (B)	-1 (C)	0 (-)	9 (B)	0 (D)	1 (D)
w	6 (B)	1 (C)	0 (-)	9 (B)	9 (D)	1 (D)

5 Extended TRSP model for different objective functions

As mentioned in the previous section, in the TRSP, the objective function is defined in terms of costs assigned to vertices and edges of the selected cliques. Since the train routes that belong to the selected cliques are used in the rtRTMP solution, we study whether, and to which extent, customization of the TRSP cost computation is necessary when changing the objective function used in the rtRTMP. In particular, we evaluate the correspondence between the TRSP and rtRTMP objective functions to identify the routes that lead to the best rtRTMP solutions. Therefore, in this section, we propose for each rtRTMP objective function presented in Section 3.1.1 its formalization in the TRSP.

To approximate as closely as possible the rtRTMP objective functions, we also extend the cost estimation model. Specifically, we introduce new edge costs to account for delays at different points in the control area, i.e., entry and exit points. Remark that the edge cost w_{ij} proposed by Pascariu et al. [22] (described in Section 4) estimates the overall additional running time along the potential waiting train’s route when v_i and v_j , linked by e_{ij} , are simultaneously considered. In this paper, we make a further step to assess the potential deviation from the timetable entry and exit time due to conflicts between train route pairs. To do so, we define the potential entry delay w_{ij}^{in} and the potential exit delay w_{ij}^{out} . They are edge costs that can be considered either separately or concurrently. These edge costs are derived from the estimation of w_{ij} in Section 4 as follows.

The potential entry delay w_{ij}^{in} is defined in Eq. (3) as the minimum between the utilization overlaps computed for the two train routes v_i, v_j on the common section(s) res^{in} which include their entry point (if any). This cost is computed during the fixed-component calculation. That is when the utilization overlap on the common section(s) is estimated as the difference between the end of the utilization by the first passing train $eU_{v_{i[j]}}$ and the start of the utilization by the second one $sU_{v_{j[i]}}$.

$$w_{ij}^{\text{in}} = \min(eU_{v_i} - sU_{v_j}, eU_{v_j} - sU_{v_i}) \quad \forall v_i, v_j \in V : \exists res^{\text{in}} \quad (3)$$

For the computation of the potential exit delay w_{ij}^{out} , let us consider the general cost w_{ij} (given by train routes v_i, v_j) and the train t , using route v_i , as the potential waiting train. We compute w_{ij}^{out} based on the fixed component for clique constructions as in Eq. (4). We assume that, if v_i has running time (r_i) equal to or longer than the timetable route v_d (r_d) in a free network, the additional running time due to the interaction of both train routes represented by w_{ij} directly

translates into exit delay. Instead, if v_i is shorter, t can absorb additional potential delay w_{ij} . We thus estimate that the potential exit delay w_{ij}^{out} will be smaller than w_{ij} by $r_d - r_i$.

$$w_{ij}^{\text{out}} = \begin{cases} w_{ij} & \text{if } r_i \geq r_d, \\ w_{ij} - (r_d - r_i) & \text{otherwise.} \end{cases} \quad (4)$$

The values of potential entry and exit delays are related in the case of timetable routes: w_{ij}^{out} can only be greater or equal to w_{ij}^{in} . Instead, when considering alternative train routes, the different minimum running times and the procedure we apply to compute w_{ij} can lead to a range of possible relations between w_{ij}^{in} and w_{ij}^{out} . For example, a large w_{ij} can be recovered if the potential waiting train uses a much faster route than the timetable one, resulting in a small or null w_{ij}^{out} . In this case, the potential entry delay w_{ij}^{in} can possibly be larger than w_{ij}^{out} if the v_i and v_j have a common section with a positive overlap at their beginning.

In the following, we propose customized formulations of the TRSP, driven by the choice of objective functions in the rtRTMP. We recall that part of the constraints for the problem have already been declared in Section 4, and presented in Constraints (1).

The *Total Potential Exit Delay* (\mathcal{P}_{TED}) (5) minimizes the aggregate potential delay that all trains might experience at the exit points of the control area. According to this formulation, the clique cost is computed as the sum of all vertex costs u_i and edge costs w_{ij}^{out} in clique c .

$$\begin{aligned} \min \quad & \sum_{v_i \in V} u_i x_i + \sum_{e_{ij} \in E} w_{ij}^{\text{out}} y_{ij} \\ \text{s.t.} \quad & \\ & \text{Constraints (1)} \end{aligned} \quad (5)$$

The *Total Potential Delay* (\mathcal{P}_{TD}) (6) is a variant of \mathcal{P}_{TED} , which also includes the minimization of potential entry train delays. The clique cost is defined in \mathcal{P}_{TD} as the sum of its vertex costs u_i , and both w_{ij}^{out} and w_{ij}^{in} edge costs.

$$\begin{aligned} \min \quad & \sum_{v_i \in V} u_i x_i + \sum_{e_{ij} \in E} (w_{ij}^{\text{out}} + w_{ij}^{\text{in}}) y_{ij} \\ \text{s.t.} \quad & \\ & \text{Constraints (1)} \end{aligned} \quad (6)$$

The *Total Potential Travel Time* (\mathcal{P}_{TTT}) (7) is the minimization of the travel time potentially spent by all trains in the control area. We estimate the total potential travel time as the sum of the minimum running time r_i of train route v_i and the general potential delay w_{ij} due to scheduling decisions along the train routes in clique c . From this value, we subtract the potential entry delay w_{ij}^{in} , since for its duration the train has not yet entered the control area. We remark that this objective function has by definition much higher values than \mathcal{P}_{TED} and \mathcal{P}_{TD} because it considers the running times typically expressed in seconds.

$$\begin{aligned} \min \quad & \sum_{v_i \in V} r_i x_i + \sum_{e_{ij} \in E} (w_{ij} - w_{ij}^{\text{in}}) y_{ij} \\ \text{s.t.} \quad & \\ & \text{Constraints (1)} \end{aligned} \quad (7)$$

The *Potential Number of Delayed Trains* (\mathcal{P}_{NDT}) (8) estimates the number of trains which are delayed at the exit from the control area. Let $\bar{I}(v_i)$ be the set of edges incident in v_i for which the train using route v_i is the potential waiting train. A train using a specific route is considered delayed if the corresponding vertex $v_i \in c$ has either a cost $u_i > 0$ or an incident edge $e_{ij} \in \bar{I}(v_i)$

with potential delay $w_{ij}^{\text{out}} > 0$. To quantify the potential number of delayed trains, we introduce the binary variable z_t , with $t = 0, \dots, k$, and a big M constant. We use it in Eqs. (8) to state whether a train is potentially delayed ($z_t = 1$) or not ($z_t = 0$) when using route v_i . The clique cost minimized is thus formulated as the sum of z_i for all $v_i \in c$. \mathcal{P}_{NDT} has by definition a very low value, whose maximum is the total number of trains. This formulation disregards the information about potential delay values, so it may lead to select cliques with few strongly delayed trains.

$$\begin{aligned}
& \min \sum_{t \in T} z_t \\
& \text{s.t.} \\
& u_i x_i + \sum_{e_{ij} \in \bar{I}(v_i)} w_{ij}^{\text{out}} y_{ij} > z_t - 1 \quad \forall t \in T, v_i \in V_t \\
& u_i x_i + \sum_{e_{ij} \in \bar{I}(v_i)} w_{ij}^{\text{out}} y_{ij} \leq M z_t \quad \forall t \in T, v_i \in V_t \\
& z_t \in \{0, 1\} \\
& \text{Constraints (1)}
\end{aligned} \tag{8}$$

The *Maximum Potential Delay* (\mathcal{P}_{MD}) (9) is the one suffered by the train exiting the control area with the maximum potential delay. We assume the most potentially delayed train is the one that maximizes the sum of the potential delay given by the route additional travel time u_i and the potential exit delays $w_{ij}^{\text{out}} \in \bar{I}(v_i)$ assigned to it. In Eqs. (9), the clique cost variable f_d represents the maximum potential exit train delay, which is minimized by the objective function.

$$\begin{aligned}
& \min f_d \\
& \text{s.t.} \\
& f_d \geq u_i x_i + \sum_{\bar{I}(v_i)} w_{ij}^{\text{out}} y_{ij} \quad \forall v_i \in V \\
& \text{Constraints (1)}
\end{aligned} \tag{9}$$

The *Maximum Potential Completion* (\mathcal{P}_{MC}) (10) estimates the potential completion time of a train route $v_i \in c$ as the sum of its timetable entry time in the control area s_i , the minimum running time r_i of train route v_i and the additional running time due to conflicts with other trains $w_{ij} \in \bar{I}(v_i)$ assigned to it. As in \mathcal{P}_{MD} , we compute the overall additional running time of a specific train route assignment $v_i \in c$ as the sum of all incident edges' cost for which the train using v_i is the potential waiting one. The clique cost f_c is thus given in Eqs. (10) by the train route assignment $v_i \in c$ which maximizes the potential completion time. Similarly to \mathcal{P}_{TTT} , the \mathcal{P}_{MC} objective value is in general high due to the train running time and entry time. The latter typically implies a significantly larger magnitude. For example, for a train entering the control area at 3 pm s_i and spending there one hour, if we measure time in seconds we will have $s_i = 54000$ and $r_i + w_{ij} = 3600$.

$$\begin{aligned}
& \min f_c \\
& \text{s.t.} \\
& f_c \geq (s_i + r_i) x_i + \sum_{\bar{I}(v_i)} w_{ij} y_{ij} \quad \forall v_i \in V \\
& \text{Constraints (1)}
\end{aligned} \tag{10}$$

Example

Based on the example in Section 4, we illustrate here the extensions to the cost estimation model introduced for the different objective functions.

The potential entry delay w^{in} is obtained as follows. We consider routes a_1 and b_2 that share their first section s_1 . The potential entry delay corresponds to the minimum utilization overlap on s_1 , i.e., $w_{a_1b_2}^{\text{in}} = \min(3, 5) = 3$. The pairs of routes a_1c_2 and a_1d_3 do not share any of the initial sections. Regarding b_2 and d_3 , due to the rolling stock re-utilization between trains B and D, these routes share necessarily the first section of train D. However, since the given process time for the re-utilization operation can be respected (see Appendix A) no potential entry delay is recorded. Instead, train routes b_2 and c_2 only share c_2 's first section s_{10} . In this case, a potential entry delay is recorded only if C is the potential waiting train. When train B passes first on this section, the utilization overlap is 17 (see Appendix A), while when C is first it is 9. Considering that the minimum value results when train C passes first and it does not have to wait to enter the infrastructure, its potential entry delay is null $w_{b_2c_2}^{\text{in}} = 0$. The potential entry delays of b_2d_3 and c_2d_3 in Table 4 are obtained by the same procedures as for a_1b_2 and b_2c_2 .

Table 4: Edge costs, in brackets the potential waiting train

	a_1b_2	a_1c_2	a_1d_3	b_2c_2	b_2d_3	c_2d_3
w^{in}	3 (B)	0 (-)	0 (-)	0 (-)	0 (-)	1 (D)
w^{out}	6 (B)	1 (C)	0 (-)	9 (B)	5 (D)	1 (D)

The potential exit delay instead is directly derived from w . As discussed in Section 5, when the train suffering the potential delay w travels on a faster route than the timetable one, the train may be able to catch up some of its delay. Thus, w is reduced by this corresponding amount. For example, the potential delay $w_{b_2d_3} = 9$ is suffered by train D, and the current route d_3 is faster than the timetable route by 4-time units. Thus, the potential exit delay results from $w_{b_2d_3}^{\text{out}} = 9 - 4 = 5$. Table 4 shows that $w^{\text{out}} = w$ for the other pairs of train routes. The reason is that the route of the potential waiting train is not faster than the timetable one, and so it does not allow for the partial recovery of w .

Based on vertex and edge cost, we now compute the clique cost according to the different objective functions:

$$\mathcal{P}_{TED} = (0 + 1 + 3 + 0) + (6 + 1 + 0 + 9 + 5 + 1) = 26 \quad (11)$$

$$\mathcal{P}_{TD} = (0 + 1 + 3 + 0) + (3 + 0 + 0 + 0 + 0 + 1) + (6 + 1 + 0 + 9 + 5 + 1) = 30 \quad (12)$$

$$\mathcal{P}_{TTT} = (21 + 20 + 24 + 16) + (3 + 0 + 0 + 9 + 9 + 0) = 102 \quad (13)$$

$$\mathcal{P}_{NDT} = 0 (z_A) + 1 (z_B) + 1 (z_C) + 1 (z_D) = 3 \quad (14)$$

$$\mathcal{P}_{MD} = \max\{(0 + 0)_{a_1}, (1 + 6 + 9)_{b_2}, (3 + 1)_{c_2}, (0 + 5 + 1)_{d_3}\} = 16 (B) \quad (15)$$

$$\mathcal{P}_{MC} = \max\{(8 + 21 + 0)_{a_1}, (9 + 20 + 6 + 9)_{b_2}, (20 + 24 + 1)_{c_2}, (29 + 16 + 9 + 1)_{d_3}\} = 55 (D) \quad (16)$$

6 pACO-TRSP algorithm

In this paper, we use Pascariu et al. [22]'s parallel pACO-TRSP algorithm to solve the TRSP. While a brief overview of this algorithm is given in this section, for a more comprehensive description we refer to the mentioned reference paper. ACO is a well-known metaheuristic inspired by the ant foraging behavior to solve hard combinatorial problems [11]. The algorithm uses a virtual ant colony to explore the solution space while laying numerical pheromones on the best solutions to drive the search toward high-quality regions of the feasible space.

pACO-TRSP iteratively produces solutions until a stopping criteria is reached, i.e., either all the available computational time limit has been used or a zero-value objective function has been found. At each iteration, each ant of the colony incrementally builds a clique $c \in \Gamma$ on the construction graph G selecting one vertex at a time via the random proportional rule, which is based on the pheromone trail and heuristic information. When all ants of the colony have built a clique, a parallel local search is performed for a given number of cliques with the best clique costs. For each of them, the local search aims to find a lower cost clique. After the local search, the best

clique in the current iteration is used to update the pheromone according to the MAX-MIN Ant System [32]. The best clique of each iteration is stored and, at the end of the computation, the overall best cliques are used to extract n alternative routes for each train.

pACO-TRSP can be used with all objective functions described in Section 5, without the need of major algorithmic changes. The objective function definition, however, requires the customization of the local search and the selection of the cliques used for depositing pheromones. This customization involves selecting the worst-cost vertex according to each objective function in Section 5. This implies that different local search moves will be performed when considering different objective functions, and pheromones will drive the research space exploration in different directions. As shown in Section 5, each objective function corresponds to a unique problem formulation, and our approach ensures that the algorithm dynamically adjusts its search strategy to accommodate these differences.

Algorithm 1: Local search

Data: Construction graph G , clique c
Result: Clique c of lower cost

```

1 begin
2    $v_{worst} = \{v_0\}$ 
3    $f(v_{worst}) = bigInteger$ 
4   forall  $v_i \in c$  do
5      $f(v_i) = f(c) - f(c \setminus v_i)$ 
6     if  $f(v_i) > f(v_{worst})$  then
7        $v_{worst} = \{v_i\}$ 
8    $c_{new} = c$ 
9   forall  $v_i \in V : v_{worst}, v_i \in V_t, \exists e_{ij} \in G \forall v_j \neq v_{worst} \in c$  do
10     $c' = c \setminus \{v_{worst}\} \cup \{v_i\}$ 
11    if  $f(c') < f(c_{new})$  then
12       $c_{new} = c'$ 
13  return  $c_{new}$ 

```

For a detailed understanding of the customized local search, we present the local search in Algorithm 1. Within the local search, a critical component is the identification of the worst-cost vertex given a clique c on the construction graph G . This worst-cost vertex, denoted as v_{worst} , is the one that maximizes the difference between the cost of the clique and that of the clique without the given vertex (lines 2-9, Algorithm 1). The clique cost is defined by the objective function in use, according to its formulation in Section 5. Subsequently, the local search strives to replace the identified worst-cost vertex, v_{worst} , with another vertex selected from the same partition of the construction graph G , i.e., $v_i, v_{worst} \in V_t$, such that v_i forms a new feasible clique of lower cost, if any (lines 8-13, Algorithm 1).

7 Computational experiments

We perform extensive computational analysis on two French railway infrastructures with timetable perturbations to evaluate the effectiveness of the TRSP when the rtRTMP is solved by different objective functions and solution approaches. Moreover, the experiments serve to evaluate how the TRSP is effective on the rtRTMP as long as the objective functions considered in the second problem are reflected in the first, as proposed in Section 5.

In Section 7.1, we provide an in-depth description of the specific characteristics of these case studies and discuss the various challenges that arise when dealing with real-time rail traffic management. Moving to Section 7.2, we outline the experimental setup used to evaluate the potential of the TRSP as a preprocessing step for the rtRTMP and provide a brief overview on how the

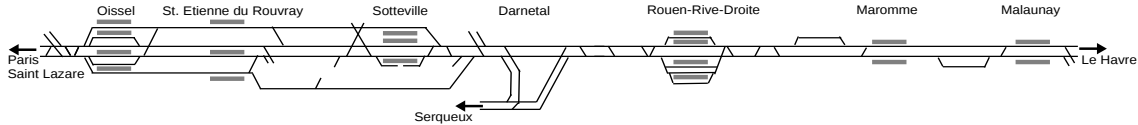


Figure 3: Rouen infrastructure

TRSP and the rtRTMP are solved using state-of-the-art solvers. Section 7.3 discusses the results and the impact of the TRSP on the rtRTMP solution, when varying the types of rtRTMP solver and the rtRTMP objective functions. Finally, Section 7.4 focuses on the relationship between the TRSP model and the objective function of the rtRTMP, evaluating the significance of their correspondence by the Wilcoxon signed rank test.

7.1 Case studies

The case studies used in the computational experiments represent two French control areas that have different characteristics: a line with intermediate stops and mixed traffic, and a passenger terminal station with high density traffic. We chose these two case studies to take into account several aspects that could influence the suitability of the TRSP application in the context of real-time rail traffic management. Specifically, these case studies offer different perspectives as they vary in the size of the railway network, encompassing different numbers and types of trains and routes. This enables us to evaluate the ability of the TRSP to effectively handle different network scales, as well as different types of railway traffic and networks. In the following, we provide a more in-depth description of the particular attributes of these case studies, and we discuss the distinct challenges that emerge when dealing with real-time railway traffic management. Table 5 details an overview of the case study characteristics. Columns 1-2 identify the test case and report its length, columns 3-6 present respectively the number of block sections, track circuits, available routes and trains per hour running in the network, column 7 the average $[\min, \max]$ number of alternative routes per train, while columns 8-9 the average number of vertexes and edges in the TRSP model.

Table 5: Case studies characteristics

Case study	Length (km)	Block sections	Track circuits	Routes	Trains /hour	Routes /train	TRSP	
							$ V $	$ E $
Rouen	27	176	190	1744	12	42[1,192]	463	91298
Lille	12	829	299	2409	41	163[1,458]	6623	21879902

Figure 3 shows the first test case (referred to as Rouen): a 27-km railway line, connecting Paris Saint Lazare, Serqueux and Le Havre through the Rouen junction. A three-aspect signaling system is deployed on this line. The infrastructure in this control area is mainly a double-track line including six stations and only a few possibilities for changing tracks between stations. The presence of the rather large stations at Oissel, St. Etienne du Rouvray, Sotteville and Rouen-Rive-Droite, leads some trains to have up to 192 alternative routes, which can make the rerouting problem challenging. Additionally, the mixed traffic, including freight, local, and high-speed trains, poses another challenge due to the significant variations in running times between different types of trains.

Figure 4 shows the second test case (referred to as Lille) including the Lille-Flandres terminal station area. Lille is a major terminal station with a topology articulated in 17 platforms that branch in national and international lines used by local, intercity, and high speed trains. The area implements a three-aspect signaling system. Despite its relatively short length of 12 kilometers, this control area presents an interesting challenge due to the intricate and overlapping nature of train routes. Most of the trains have the possibility of traversing a significant part of the

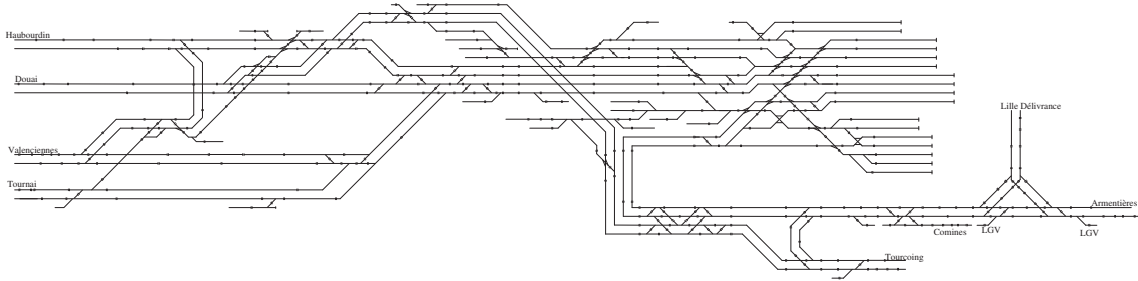


Figure 4: Lille infrastructure

infrastructure by changing their route. As shown in Figure 4, the infrastructure layout allows trains to independently access nearly any of the 17 platforms, irrespective of their origin or destination lines. Often, multiple options are available for connecting a platform to a line track. As a result, a very large number of alternative routes are available for each train, with up to 458 alternative routes per train. These alternative routes may appear almost identical in the early half of their part connecting to the open line. However, they have significant variations as they approach the platforms. Being a terminal station, trains frequently encounter other trains traveling in the opposite direction. Furthermore, all trains must turnaround at platforms, so there are constraints on rolling stock re-utilization between pairs of trains. This adds complexity to traffic management, since decisions affecting several trains may be mutually related, even if there are no direct conflicts between these trains.

For each case study, we consider 50 perturbed scenarios of one-hour traffic. These perturbed scenarios are generated as follows.

We consider an actual one-day timetable. We perturb this initial timetable with a train delay between 5 and 15 minutes affecting the entry time of 20% of trains, randomly selected [23, 25]. We thus generate a total of 5 perturbed one-day timetables. For each perturbed timetable, we draw ten 60-minute time windows, randomly selected during the 7:30-9:00 and 18:30-20:00 peak hours.

In the case of Rouen, 186 trains run on the one-day timetable. Whereas, in the 50 perturbed scenarios, an average of 12 trains are running per hour. Instead, the Lille station area is traversed by 509 trains in the one-day timetable and there are on average 41 trains running per hour.

The resulting TRSP construction graph G has an average number of vertices $|V| = 463$ and edges $|E| = 91\,298$ for Rouen, and $|V| = 6\,623$ and $|E| = 21\,879\,902$ for Lille.

7.2 Experimental settings

We assess the potential of the TRSP as a preprocessing step when the rtRTMP is solved with different objective functions and by different solvers, each having a specific model and solution approach. Specifically, we solve the TRSP model formulated in Sections 4 and 5 with a computational time limit of 30 seconds. Subsequently, we tackle the rtRTMP using the alternative train routes belonging to the TRSP solution, with a computational time limit of 150 seconds. For the rtRTMP solution, we consider two state-of-the-art solvers: RECIFE-MILP [24] and AGLIBRARY [29].

To assess the performance of our TRSP-rtRTMP approach, we compare it with the state-of-the-art for these solvers. In this comparison, we consider the default timetable route for each train and all available alternative routes. We set a computational time limit of 180 seconds for the rtRTMP solution process. We call this reference approach *no-TRSP*. We chose a computational time of 180 seconds aligning with recommendations from infrastructure managers, as detailed in Quaglietta et al. [26]. Typically, the rtRTMP solution is part of a larger and context-aware decision support framework for dispatchers, such as closedloop systems [1, 8]. Since rail traffic varies rapidly in real time, the rtRTMP must respond quickly to disturbances in order to propose appropriate solutions. For this purpose, the problem can be solved periodically or triggered by the detection of a disturbance [33]. In this context, the rtRTMP proactively manages potential deviations from the

timetable to maintain constant operational efficiency. Considering such a bigger framework, the computational time of 180 seconds is compatible if train traffic is managed proactively, considering a delay propagation forecast.

In the following, we provide a brief overview of the rtRTMP solvers used. For a comprehensive description, we refer the readers to the original papers that introduced these methods.

RECIFE-MILP [24]. RECIFE-MILP models the rtRTMP as described in [24], considering simultaneously the variables of timing, ordering and train routing related to the decisions that dispatchers can make by monitoring a specific control area. The time variables are continuous and represent the beginning and ending times of track circuit utilization, delay, and physical occupancy. Train ordering and routing variables are binary variables. Big-M constraints are used to manage train scheduling variables in RECIFE-MILP. These constraints enforce the capacity of the network and model the route-lock sectional-release interlocking system, in which a track circuit can only be used by one train at a time, except when the train is engaged in a rolling stock re-utilization operation on the track circuit. RECIFE-MILP solves the MILP formulation via a commercial MILP solver, which implements performance-boosting methods for real-time applications with a limited computational time [24]. As such, it can optimize virtually any linear objective function. Among the objective functions described in Subsection 3.1.1, we consider the minimization of: TED, TD, TTT, and NDT.

AGLIBRARY [29]. AGLIBRARY models the rtRTMP as a Flexible Job Shop Scheduling Problem using the alternative graph formulation of Mascis & Pacciarelli [18]. The model considers a train as a job and a track circuit as a resource, enabling the definition of an operation o as the process of a train traveling along a track circuit (i.e., the use of a resource by a job) when using one of its available routes. AGLIBRARY solves the problem iteratively: at first, in a rescheduling module, an initial scheduling problem with default routes is solved by the branch and bound algorithm of [9]; then, in a rerouting module, the incumbent solution is evaluated to select, among the available ones, new routes for the more critical trains, using a variable neighborhood search metaheuristic. The new routes are then passed again to the rescheduling module to obtain an improved solution. The process goes on until either the available computation time has elapsed or a zero delay solution is obtained [29]. AGLIBRARY minimizes makespan-related objective functions exploiting the alternative graph model. Thus, among the objective functions described in Subsection 3.1.1, we consider the minimization of: MC and MD.

We carry out a full factorial computational analysis, which addresses all possible combinations of TRSP and rtRTMP objective functions. We perform a total of 2100 experiments for each case study: for each of the six rtRTMP objective functions in Section 3.1.1, we solve the problem for the 50 perturbed scenarios with no TRSP and with the TRSP approach using each of the six objective functions in Section 5. We refer to the objective functions used in the TRSP after the acronyms in Section 5: \mathcal{P}_{TED} , \mathcal{P}_{TD} , \mathcal{P}_{TTT} , \mathcal{P}_{NDT} , \mathcal{P}_{MD} and \mathcal{P}_{MC} .

Following previous studies and preliminary computational analysis, we consider a different cardinality of TRSP solutions (i.e., the number of train route alternatives to select and provide in input to the rtRTMP) for the two case studies: 30 route alternatives for the Rouen case study [28], and 10 for the Lille case study [22]. This difference is due to the size of the instances and the ability of the rtRTMP to have good performance using either more or fewer routes.

The experiments are performed in a laboratory environment. RECIFE-MILP is run on a workstation Intel Xeon 16 core 3.5 GHz processor with 128 GB RAM, under Linux Ubuntu distribution, using CPLEX 12.6 as MILP solver. AGLIBRARY is run on a workstation Intel Xeon 22 core 2.2 GHz processor with 1.5 TB RAM, under Windows distribution. The pACO-TRSP algorithm is implemented in C++ and it is run on the respective system of the rtRTMP approach using 16 threads.

In Section 7.3, we assess the impact of the TRSP on the rtRTMP solution when varying the types of rtRTMP solver (model and solution process) and the rtRTMP objective functions. Then in Section 7.4, we evaluate how much the performance is impacted by whether or not the TRSP matches the rtRTMP objective function.

Table 6: Results of rtRTMP performance for Rouen instances. Objective functions minimized in the rtRTMP: total exit delay (TED), total delay (TD), total travel time (TTT), number of delayed trains (NDT), maximum delay (MD), maximum completion (MC). The computational time and objective value are expressed in seconds, except for the NDT objective value.

TRSP	RECIFE-MILP									AGLIBRARY								
	TED			TD			TTT			NDT			MD			MC		
Approach	obj	gap	time	obj	gap	time	obj	gap	time	obj	gap	time	obj	gap	time	obj	gap	time
<i>no-TRSP</i>	893	63	129	1047	65	131	11617	0.00	50	1.68	48	115	102	63	156	54383	0.07	180
\mathcal{P}_{TED}	245	58	95	284	60	94	11623	0.05	45	1.02	47	70	61	49	124	54376	0.05	180
\mathcal{P}_{TD}	245	60	98	281	59	94	11624	0.06	44	1.06	51	70	61	49	131	54378	0.05	180
\mathcal{P}_{TTT}	252	63	96	288	63	94	11622	0.08	44	1.08	50	68	72	51	180	54379	0.07	180
\mathcal{P}_{NDT}	250	59	105	286	58	95	11624	0.05	45	1.00	46	86	77	52	125	54377	0.05	180
\mathcal{P}_{MD}	271	61	93	345	61	98	11624	0.06	44	1.10	50	79	61	49	124	54376	0.05	180
\mathcal{P}_{MC}	311	60	112	356	61	107	11621	0.03	45	1.14	48	98	95	56	139	54375	0.05	180

22

Table 7: Results of rtRTMP performance for Lille instances. Objective functions minimized in the rtRTMP: total exit delay (TED), total delay (TD), total travel time (TTT), number of delayed trains (NDT), maximum delay (MD), maximum completion (MC). The computational time and objective value are expressed in seconds, except for the NDT objective value.

TRSP	RECIFE-MILP									AGLIBRARY								
	TED			TD			TTT			NDT			MD			MC		
Approach	obj	gap	time	obj	gap	time	obj	gap	time	obj	gap	time	obj	gap	time	obj	gap	time
<i>no-TRSP</i>	1390	100	180	2019	100	180	13748	12.09	180	6.00	100	180	393	91	177	53647	1.64	180
\mathcal{P}_{TED}	378	88	160	681	96	159	12481	1.75	51	4.88	98	164	357	85	168	53644	1.63	180
\mathcal{P}_{TD}	380	90	160	568	92	155	12474	1.70	51	5.14	94	163	371	89	175	53646	1.64	180
\mathcal{P}_{TTT}	430	90	162	708	96	160	12473	1.70	52	5.62	92	165	370	87	175	53643	1.63	180
\mathcal{P}_{NDT}	589	88	154	739	92	156	12491	1.84	50	4.00	92	165	378	82	166	53652	1.65	180
\mathcal{P}_{MD}	511	92	169	736	94	166	12494	1.86	53	6.68	98	162	355	81	163	53647	1.64	180
\mathcal{P}_{MC}	627	96	173	1174	96	170	12495	1.87	52	6.44	98	169	415	91	180	53638	1.62	180

7.3 Analysis of TRSP-rtRTMP results

Tables 6 and 7 show the computational results for all TRSP-rtRTMP combinations considered in our assessment, respectively, for Rouen and Lille instances. Column 1 specifies the TRSP approach: either *no-TRSP* or pACO-TRSP. For pACO-TRSP, the used objective function is specified. The remaining columns show the results for each rtRTMP objective function, in terms of average objective value (*obj*), optimality gap (*gap*) and computation time (*time*). When a TRSP approach is considered, the reported time also includes the computational time consumed by the pACO-TRSP algorithm. The computational time and objective value are expressed in seconds, except for the NDT objective value. The percentage optimality gap is computed as $(\text{best integer} - \text{lower bound}) / (\text{best integer})$. We consider the lower bound obtained with *no-TRSP*, respectively, by CPLEX when using RECIFE-MILP, and calculated according to D’Ariano et al. [9] when using AGLIBRARY. When the optimality gap is zero, the best lower bound has been reached. In contrast, when the optimality gap is 100%, the lower bound is equal to zero and is different from the best integer. In general, a lower optimality gap is better, as it indicates that the current solution is closer to the optimal solution. Higher optimality gaps suggest that there is still significant room for improvement in the solution quality. In Tables 6 and 7, the best result for each rtRTMP objective function is highlighted in bold.

The results in Table 6 indicate that in situations where the number of routes does not pose a significant challenge for the standalone rtRTMP solver, i.e., when the solvers are able to achieve optimality or near-optimality with all routes, the TRSP may not be necessary. In these cases, the TRSP may exclude some optimal rtRTMP solutions by limiting the number of routes, but it is still able to select routes that lead to near-optimal rtRTMP solutions. For example, the use of the TRSP does not help in certain instances of the TTT in the Rouen case study, as RECIFE-MILP is able to find optimal solutions using all routes in a short computational time. However, in instances of MC, where AGLIBRARY achieves near-optimal solutions with *no-TRSP* (with an optimality gap below 1%), pACO-TRSP is effective in finding even higher quality solutions and reducing the optimality gap further. Overall, these results indicate that the application of TRSP does not have a significant negative impact on performance when it is not needed.

Instead, when the number of possible train routes hinders the standalone solvers (i.e., *no-TRSP*) from finding the optimal solution, the TRSP always improves the rtRTMP solution. The improvement is remarkable irrespective of the rtRTMP model, the solution approach, and the optimized objective function. That is, for all objective functions in Tables 6 and 7 except those discussed above. This is particularly true for difficult instances such as the Lille ones, where dealing with a large number of routing possibilities can be computationally intensive for both RECIFE-MILP and AGLIBRARY. As highlighted in Section 7.1, in Lille, routes are very articulated and often overlapping, which necessitates their decomposition into a substantial number of track circuits and block sections, with multiple options available for connecting platforms to line tracks. In the case of RECIFE-MILP, the large number of routing possibilities translates into a huge number of binary variables and thus a highly computationally consuming exploration of the branch and bound tree by the exact algorithm. As an example, in the Lille instances, *no-TRSP* needs to deal with 24760 binary variables on average. In the case of AGLIBRARY, the metaheuristic algorithm faces the risk of getting stuck in low-quality local minima: finding good train route combinations to escape local minima becomes more difficult when all routes are considered. For both rtRTMP solvers, the accurate selection of a limited number of train routes leads to more successful exploration of the solution space, with higher quality results.

RECIFE-MILP often shows weak lower bounds close to zero for TED and TD. For Rouen, the lower bound for TED and TD is null in 42 and 45 instances, respectively. Similarly, in the Lille case study, the lower bounds for TED and TD remain consistently at zero, resulting in a 100% optimality gap when using *no-TRSP*. When the lower bound remains zero, either the solver is able to drive the problem to an optimal solution of value zero, or the optimality gap remains fixed to 100% for whatever value of the objective function. The former case occurs when the best solution found does not involve any secondary delay, while the latter occurs when the objective function has a value greater than zero. These weak lower bounds have an interesting impact on the results in Tables 6 and 7. For example, for TED, *no-TRSP* reaches the optimum for 16 instances

in a computational time shorter than 180 seconds. This lowers the average computational time. However, for the remaining 34 instances where the optimum is not reached, the optimality gap remains high, often equal to 100%. Furthermore, when comparing different TRSP approaches for the same rtRTMP objective function, those having more solutions with zero delay can show a smaller gap in optimality. At the same time, they can end up with a higher average objective value because the rest of the solutions have delays much greater than zero. An example is the result of \mathcal{P}_{NDT} for TD in Rouen case: \mathcal{P}_{NDT} displays a worse average objective value compared to \mathcal{P}_{TD} , yet its optimality gap is smaller due to having two more zero-delay solutions compared to \mathcal{P}_{TD} .

The objective values in Tables 6 and 7 reveal variations in performance between the two case studies. This variation highlights the importance of considering the specific characteristics of the case study when determining the most appropriate objective function by the stakeholders. In Rouen, the results in Table 6 show a small variation between TED and TD objective values, with an average of 50 seconds, for all TRSP approaches. This indicates that train entry delays account for a small part of the total delay. In this case study, the small number of trains entering the control area per hour implies that an average of 2 out of 12 trains have a primary delay at the entry in the control area, in the experimental setup we consider. Thus, potential conflicts and secondary delays at the entry points are rather sparse. Moreover, the infrastructure layout, which consists of open lines and intermediate stops with planned buffer times, allows the separation of railway traffic flows. This implies that the propagation of train delays can be more easily contained, resulting in reduced NDT values. *No-TRSP* provides an average of about two delayed trains, and the TRSP is able to lead to an average of about one train late. Given the low number of delayed trains, in 10 cases out of 50, RECIFE-MILP *no-TRSP* is able to find optimal solutions for all objective functions. This brings the average computational times in Table 6 below 180 seconds. However, for the non-optimal solutions, the objective values found by *no-TRSP* are significantly worse than when solving the TRSP. The small number of delayed trains in Rouen enhances also the AGLIBRARY performance for MD. The train with the maximum delay has little interference with other trains, leading to objective values that are generally low even with *no-TRSP*. The TRSP leads to better performance, and with \mathcal{P}_{MD} the maximum train delay is reduced by 40% in half of the computational time.

In the Lille case study, the number of trains entering the control area, as well as those affected by primary delays at the entry, is four times bigger than in Rouen. Thus secondary train delays are more likely to occur at the entry in the control area, as evident by the remarkable increase of TD objective values over the TED ones in Table 7. In addition, the infrastructure layout also makes it more difficult to contain the propagation of delays, resulting in a higher number of delayed trains. *No-TRSP* records an average of 6 delayed trains, and only \mathcal{P}_{NDT} achieves the best average result of 4 delayed trains. The intricate and busy network of Lille does not allow the separation of the traffic flow, thus a train is subject to a long delay because of conflicts with several trains simultaneously, affecting also MD objective values. Limiting the number of routes by \mathcal{P}_{MD} helps improve the performance of *no-TRSP* most of all, reducing average objective value by 10%. The minor improvement compared to Rouen is due to the greater difficulty in finding good train route combinations also by the pACO-TRSP algorithm. Specifically, the correspondent objective function \mathcal{P}_{MD} proves to be the best objective function for approximating MD solutions, but further customization of the TRSP model and solution structure may be needed to better assess the potential worst train.

7.4 Impact of different TRSP objective functions on rtRTMP performance

In this section, we analyze the relationship between the TRSP model and the objective function of the rtRTMP. In the previous section, Tables 6 and 7 show that aligning the TRSP with the rtRTMP objective function leads consistently to the best rtRTMP performance.

However, the magnitude of this difference may vary depending on the specific objective function and case study. To determine the statistical significance of these differences, we employ the Wilcoxon signed rank test to compare the differences in the rtRTMP solution values obtained

by each TRSP objective function and the corresponding objective function, as presented in Tables 8 and 9. In particular, from the third column onwards in the tables, each cell compares two different objective functions used in the TRSP. The value in each cell compares the rtRTMP values obtained using the objective function listed in the column (corresponding to the rtRTMP) with those obtained using the objective function listed in the row. For the Wilcoxon signed rank test, we assume a confidence level of 0.99. The null hypothesis of the test assumes no difference between the objective functions, i.e., the pseudo-median μ of the difference is equal to zero. Positive values of μ and of the lower and upper bounds of the confidence interval (CI) mean that the difference is significant and the correspondent objective function performs significantly better. Vice-versa, negative values of μ and the lower and upper bounds indicate that the solutions of the non-corresponding objective function are significantly better. When the lower and upper bounds have opposite signs, the observed difference between the two objective functions is not significant. The p -value is an additional indicator of the test related to the significance level of the sample. It indicates how likely the observed data would be obtained if the null hypothesis were true. A p -value smaller than $1 - \text{confidence level}$ means stronger evidence in favor of the alternative hypothesis. In Tables 8 and 9, we highlight in grey when the difference between the objective functions is statistically significant with a confidence level of 0.99.

Table 8: Wilcoxon sign-rank test of the difference between TRSP objective functions for rtRTMP solution, in Rouen instances , with a confidence level of 0.99.

TRSP	statistics	rtRTMP					
		TED	TD	TTT	NDT	MD	MC
\mathcal{P}_{TED}	μ		17	2	1	3	7
	CI	-	{-50;71}	{-17;18}	{-2;3}	{-27;37}	{7;7}
	p -value		0.27	0.71	0.59	1.00	0.10
\mathcal{P}_{TD}	μ	1		5	0	9	22
	CI	{-41;49}	-	{-15;19}	{-1;2}	{-25;43}	{-2;79}
	p -value	0.98		0.57	0.82	0.71	0.02
\mathcal{P}_{TTT}	μ	5	9		0	41	19
	CI	{-55;83}	{-38;55}	-	{0;1}	{5;69}	{-22;75}
	p -value	0.57	0.42		0.48	0.005	0.03
\mathcal{P}_{NDT}	μ	0	2	4		43	1
	CI	{-63;125}	{-43;67}	{-8;18}	-	{18;61}	{-14;63}
	p -value	0.43	0.80	0.40		0.0004	0.93
\mathcal{P}_{MD}	μ	25	23	4	0		9
	CI	{-13;199}	{-2;280}	{-17;23}	{-1;2}	-	{-1;12}
	p -value	0.24	0.12	0.60	0.61		0.40
\mathcal{P}_{MC}	μ	55	45	5	0	86	
	CI	{0;220}	{3;139}	{-11;27}	{-1;2}	{42;137}	-
	p -value	0.06	0.02	0.20	0.32	0.0001	

In the Rouen case study, Table 8 shows that the pseudo-median is consistently positive, with only five occurrences of null values. This trend suggests a preference for the corresponding objective function. However, the confidence intervals often display opposing signs, indicating that the differences between the corresponding objective function and other alternatives lack statistical significance at a confidence level of 0.99. This indicates that while using the corresponding objective function is generally advantageous, it may not be strictly necessary. Additionally, the higher p -values compared to those in Table 9 for Lille confirm that the difference between objective values is generally small. It is worth noting that if we decrease the level of confidence, some differences in objective functions may become statistically significant. For example, \mathcal{P}_{MC} is not directly tied to train delays as it aims to minimize the last potential train arrival, and thus results in significantly different performance from the other objective functions at a lower level.

Looking at the columns TTT, NDT and MC in Table 8 no significant difference occurs between

Table 9: Wilcoxon sign-rank test of the difference between TRSP objective functions for rtRTMP solution, in Lille instances , with a confidence level of 0.99.

TRSP	statistics	rtRTMP					
		TED	TD	TTT	NDT	MD	MC
\mathcal{P}_{TED}	μ		47	9	1	8	5
	CI	-	{-74;276}	{-7;27}	{0;2}	{1;13}	{4;8}
	p -value		0.25	0.02	0.02	0.002	0.000000051
\mathcal{P}_{TD}	μ	0		6	0	10	5
	CI	{-69;80}	-	{-2;16}	{0;3}	{4;20}	{4;8}
	p -value	0.10		0.11	0.56	0.000026	0.000000023
\mathcal{P}_{TTT}	μ	16	112		2	10	4
	CI	{-7;103}	{5;265}	-	{1;3}	{2;26}	{2;7}
	p -value	0.22	0.006		0.0003	0.003	0.00000005
\mathcal{P}_{NDT}	μ	95	183	18		11	8
	CI	{-3;173}	{18;340}	{2;37}	-	{4;25}	{5;18}
	p -value	0.004	0.002	0.0001		0.0001	0.000000003
\mathcal{P}_{MD}	μ	101	132	20	2		9
	CI	{19;210}	{39;265}	{2;42}	{2;4}	-	{5;15}
	p -value	0.002	0.001	0.00006	0.000001		0.0000001
\mathcal{P}_{MC}	μ	208	442	25	2	41	
	CI	{83;420}	{242;793}	{9;39}	{1;4}	{21;98}	-
	p -value	0.0002	0.0000002	0.0000002	0.00005	0.00000005	

any TRSP objective functions. Given the ease of finding good solutions in these cases, the train routes selected by any TRSP approach performs well.

Regarding the other rtRTMP objectives in Table 8, we observe that \mathcal{P}_{NDT} leads to good performance (small pseudo-medians) as it allows for a balanced combination of train routes from different cliques. Recall that for Rouen we select 30 alternative routes for each train when the TRSP is applied. Each route being part of a different clique. The overall selected cliques have an average of one potential train route affected by high potential delays. Thus many train routes without potential delays can be combined across the whole set of cliques, rather than using the ones from the first clique. When \mathcal{P}_{NDT} cliques are used for TED, the 40% of routes selected in the rtRTMP solutions are on average from the first clique selected by the TRSP and the remaining 60% of routes are from the next 29 cliques. Instead, when using \mathcal{P}_{TD} , \mathcal{P}_{TED} or \mathcal{P}_{TTT} , TED solutions have almost 65% of routes from the first clique. The same result applies to TD, TTT, and MC, while for MD the solver gets stuck in local minima and does not reach a high quality solution with \mathcal{P}_{NDT} .

In the Lille case study (Table 9), it is clear that aligning the TRSP model with the objective function of the rtRTMP leads to the best results. The increased complexity of the system, with more routes, trains, and secondary delays, results in a greater diversity of good solutions for different objective functions. As a result, TRSP approaches that deviate from the corresponding objective function tend to be significantly worse. Specifically, when objective functions of the type *total* (TD, TED, TTT and NDT) are optimized using TRSP objective functions of the type *maximum* (\mathcal{P}_{MD} and \mathcal{P}_{MC}), a significant difference is observed at a confidence level of 0.99. This difference arises because maximum-type objective functions prioritize minimizing the worst train performance while disregarding the performance of other trains within the control area, even if they match that of the worst-performing train. Symmetrically, when optimizing *maximum* objectives (MD, MC) using TRSP objective values of type *total* (\mathcal{P}_{TED} , \mathcal{P}_{TD} , \mathcal{P}_{TTT} , \mathcal{P}_{NDT}) the significant difference arises due to their preference of (often) strongly penalizing one or few trains with respect to the others.

Focusing on total-type objective functions, it becomes apparent that TED is influenced by entry delays, total travel time, and the number of delayed trains. Consequently, the p -value for TED

indicates a significant difference compared to other total-type objective functions at a significance level below 0.99. On the other hand, for TD, the inclusion of both entry and exit delays results in substantial differences compared to \mathcal{P}_{TTT} and \mathcal{P}_{NDT} at a confidence level of 0.99.

Based on this analysis, we can conclude that although it is always preferable to align TRSP and rtRTMP, the strengths of doing so increase with the complexity of the rtRTMP problem to be addressed.

8 Conclusions and further research

This paper investigates the potential of the TRSP for its general application to the rtRTMP. Train rerouting decisions bring one of the biggest benefits to the rtRTMP solution. At the same time, they pose a major challenge: human dispatchers hardly predict the effect of simultaneous train rerouting decisions, while in advanced technology systems routing variables exponentially increase the number of solutions (combinations of scheduling and routing decisions) to explore. The TRSP addresses this challenge by selecting a limited number of routes that can work best for the rtRTMP. In this paper, we extended the applicability of the TRSP and evaluated its general effectiveness for the rtRTMP. We extend the TRSP model to consider a range of objectives often used for the rtRTMP: we propose the TRSP formulation for six objective functions and extend the cost estimation model to evaluate the correspondence between the TRSP and rtRTMP objective functions. Furthermore, we broadened the applicability by considering two different solvers, whose models and solutions approaches differ. An extended analysis has been carried out on two French networks to investigate the contribution of the TRSP in each configuration of the rtRTMP.

The results provide the following insights on the TRSP and the practical implications of its application in decision support systems for the real-time railway traffic management:

- When the number of train routes becomes a computational bottleneck, the TRSP proves to be a valuable tool in optimizing rail traffic management. Instead, when the number of available train routes does not significantly challenge the standalone rtRTMP solver (e.g., when *no-TRSP* can achieve optimality or near-optimality) and the inclusion of the TRSP may not be necessary, still, it consistently selects routes that lead to near-optimal rtRTMP solutions. Thus, infrastructure managers should consider implementing the TRSP as a strategic pre-processing in their decision support systems for addressing computational complexity and improving rail traffic management efficiency. Indeed, the results show that the TRSP is a reliable resource for improving operational outcomes in a variety of rail network settings;
- The accurate selection of a limited number of train routes by the TRSP leads to a more successful exploration of the rtRTMP solution space for both solvers: the TRSP improves RECIFE-MILP performance up to 73% when the objective function concerns the performance of all trains in the control area, and it improves the ones of AGILIBRARY of up to 40% when minimizing the worst train performance. Thus, infrastructure managers can benefit from the versatility of the TRSP and apply it to different decision support systems. Regardless of the specific rtRTMP solver used, the characteristics of the rtRTMP model, its solution approach, or the chosen objective function, the TRSP consistently leads to improved rtRTMP solutions;
- The analysis of the relationship between the TRSP model and the objective function of the rtRTMP underlines the need to adapt TRSP formulations to the different objectives of the rtRTMP. While utilizing the corresponding objective function consistently proves advantageous, its strict necessity varies depending on the type of objectives and characteristics of the infrastructure. When primary delays cause a small negative impact on train operations, as in the Rouen case study, train-oriented objective functions are more correlated with each other, i.e., the difference between the correspondent objective function and other objective functions is not always significant. In contrast, a large number of routes, trains, and secondary delays, as in the case of Lille, causes more diversification among the good solutions for different objective functions. In this case, aligning TRSP and rtRTMP objective functions always leads

to significantly better solutions. Therefore, to obtain the best results, regardless of the case study, it is recommended to correlate the TRSP model with the rtRTMP objective function.

To further improve the performance of the TRSP when changing the rtRTMP objective function, further customization of the TRSP model and algorithm might be considered. Indeed, our results suggest that the objective functions of type maximum (i.e., those seeking the worst train performance among all trains in the control area), compared to the ones of type total, change how the TRSP solution space is and should be explored. Future research will be dedicated to developing a more effective TRSP solution process for these different cases. Moreover, according to the actual TRSP model, the route choice is based on the given traffic situation and is highly dependent on the precision with which the cost estimations can model it. Future research may study a more robust train route selection model to identify good routes independently from the timetable.

Furthermore, we aim to explore the integration of train-specific weights in the objectives, allowing us to capture diverse characteristics and priorities inherent to individual trains, thereby enhancing the model's precision in reflecting the rtRTMP. Additionally, we plan to advance the TRSP concurrently with rtRTMP models to incorporate additional passenger-satisfying objectives.

Regarding the rtRTMP, future research could deepen the investigation of how different objectives interact in railway traffic management, identifying conflicts or synergies that could refine objectives and streamline decision-making. Moreover, exploring the practicality of multi-objective approaches in real-time railway traffic management could provide valuable insights for implementation.

References

- [1] David, B., Pascariu, B., Pellegrini, P., & Marlière, G. (2023, June). Analysis of the robustness of railway traffic management to driving behaviour noise. In 2023 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS) (pp. 1-6). IEEE.
- [2] Bettinelli, A., Santini, A., Vigo, D., 2017. A real-time conflict solution algorithm for the train rescheduling problem. *Transportation Research Part B*, 106, 237–265.
- [3] 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, 15–37.
- [4] 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.
- [5] 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 & Operations Research*, 39 (11), 2578–2593.
- [6] Caimi, G., Kroon, L., & Liebchen, C., 2017. Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, 6(4), 285-312.
- [7] 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.
- [8] Corman, F., & Meng, L. (2014). A review of online dynamic models and algorithms for railway traffic management. *IEEE Transactions on Intelligent Transportation Systems*, 16(3), 1274-1284.
- [9] 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.
- [10] Delorme, X., Gandibleux, X., & Rodriguez, J., 2009. Stability evaluation of a railway timetable at station level. *European Journal of Operational Research*, , 195(3), 780-790.
- [11] Dorigo, M., Stützle, T., 2004. Ant Colony Optimization, *MIT Press, Massachusetts Institute of Technology, Cambridge*.
- [12] Fischetti, M., & Monaci, M., 2017. Using a general-purpose mixed-integer linear program-

- ming solver for the practical solution of real-time train rescheduling. *European Journal of Operational Research*, 263(1), 258-264.
- [13] Goverde, R. M. (2007). Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B*, 41(2), 179-201.
- [14] Gholami, O., & Törnquist Krasemann, J., 2018. A heuristic approach to solving the train traffic re-scheduling problem in real-time. *Algorithms*, 11(4), 55.
- [15] Hansen, I.A., Pachl, J., 2014. Railway timetabling & operations. Hamburg: Eurailpress.
- [16] Kroon, L. G., Romeijn, H. E., Zwaneveld, P. J., 1997. Routing trains through railway stations: complexity issues. *European Journal of Operational Research*, 98(3), 485-498.
- [17] Lu, G., Ning, J., Liu, X., Nie, Y., 2022. Train platforming and rescheduling with flexible interlocking mechanisms: An aggregate approach, *Transportation Research Part E*, 159, 102622.
- [18] Mascis, A., Pacciarelli, D., 2002. Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, , 143 (3), 498-517.
- [19] 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.
- [20] Murali, P., Dessouky, M., Ordóñez, F., Palmer, K., 2010. A delay estimation technique for single and double-track railroads. *Transportation Research Part E*, 46(4), 483-495.
- [21] Pascariu, B., Samà, M., Pellegrini, P., D’Ariano, A., Pacciarelli, D., Rodriguez, J., 2021. Train routing selection problem: Ant colony optimization versus integer linear programming. *IFAC-PapersOnLine*, 54(2), 167-172.
- [22] Pascariu, B., Samà, M., Pellegrini, P., D’Ariano, A., Rodriguez, J., & Pacciarelli, D., 2022. Effective train routing selection for real-time traffic management: Improved model and ACO parallel computing. *Computers & Operations Research*, 105859.
- [23] 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.
- [24] 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 Trans. on Intelligent Transportation Systems*, , 16 (5), 2609-2619.
- [25] Pellegrini, P., Pesenti, R., Rodriguez, J., 2019. Efficient train re-routing and rescheduling: Valid inequalities and reformulation of RECIFE-MILP. *Transportation Research Part B*, 120 (1), 33-48.
- [26] Quaglietta, E., Pellegrini, P., Goverde, R.M.P., Albrecht, T., Jaekel, B., Marlière, G., Rodriguez, J., Dollevoet, T., Ambrogio, B., Carcasole, D., Giaroli, M., Nicholson, G., (2016). The ON-TIME real-time railway traffic management framework: A proof-of-concept using a scalable standardised data communication architecture. *Transportation Research Part C*, 63 (1), 23-50.
- [27] Samà, M., Meloni, C., D’Ariano, A., Corman, F., 2015. A multi-criteria decision support methodology for real-time train scheduling. *Journal of Rail Transport Planning & Management*, 5(3), 146-162.
- [28] Samà, M., Pellegrini, P., D’Ariano, A., Rodriguez, J., Pacciarelli, D., 2016. Ant colony optimization for the real-time train routing selection problem. *Transportation Research Part B*, 85 (1), 89-108.
- [29] Samà, M., D’Ariano, A., Corman, F., Pacciarelli, D. 2017a. A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations. *Computers & Operations Research*, 78, 480-499.
- [30] Samà, M., Pellegrini, P., D’Ariano, A., Rodriguez, J., Pacciarelli, D., 2017b. On the tactical and operational train routing selection problem. *Transportation Research Part C*, 76, 1-15.
- [31] Solnon, C., Bridge, D., 2006. An Ant Colony Optimization Meta-Heuristic for Subset Selection Problems. In *System Engineering using Particle Swarm Optimization*. Nova Science publisher, 7-29.
- [32] Stützle, T., Hoos, H.H., 2000. MAX-MIN ant system. *Future Generation Computer Systems*, 16 (8), 889-914.

- [33] Tessitore, M.L. , Samà, M., D’Ariano, A., H elouet, L., Pacciarelli, D., 2022. A simulation-optimization framework for traffic disturbance recovery in metro systems. *Transportation Research Part C*, 136, 103525.
- [34] Toletti, A., Laumanns, M., Weidmann, U., 2020. Coordinated railway traffic rescheduling with the Resource Conflict Graph model. *Journal of Rail Transport Planning & Management*, 15, 100173.
- [35] T ornquist , J., Persson, J., 2007. N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B*, 41(3), 342-362.
- [36] Van Thielen, S., Corman, F., Vansteenwegen, P., 2018. Considering a dynamic impact zone for real-time railway traffic management. *Transportation Research Part B*, 111 (1), 39–59.
- [37] Zhang, C., Gao, Y., Cacchiani, V., Yang, L., & Gao, Z., 2023. Train rescheduling for large-scale disruptions in a large-scale railway network. *Transportation Research Part B*, 174, 102786.
- [38] Yuan, Y., Li, S., Yang, L., Gao, Z., 2022. Real-time optimization of train regulation and passenger flow control for urban rail transit network under frequent disturbances. *Transportation Research Part E*, 168, 102942.

Appendix A

In this section, we report complementary information to the example in Section 4. Specifically, we describe the computation of the costs $w_{b_2c_2}^f = 9$ (B), $w_{b_2d_3}^f = 0$ (D), $w_{c_2d_3}^f = 1$ (D) corresponding to the pair of alternative routes considered in the example (Figure A.1).

For each selected train route, we report the list of sections crossed and the time values at which each train starts utilizing each section and exits (*out*) the control area in the perturbed scenario, disregarding the conflicts:

- a_1 (pink line): $s_1=8, s_3=12, s_5=16, s_7=23, s_9=28, out=29$;
- b_2 (blue line): $s_1=9, s_3=13, s_5=17, s_7=23, s_{10}=26, out=29$;
- c_2 (green line): $s_{10}=20, s_7=27, s_6=33, s_4=41, s_2=43, out=44$;
- d_3 (red line): $s_{10}=32, s_8=37, s_6=40, s_4=45, s_2=46, out=48$.

Remark that the time at which each train starts the utilization of its first section corresponds to the train entry time. We recall that due to the turnaround between train B and D, s_{10} cannot be occupied by another train between the end of utilization of B and the beginning of utilization of D. Moreover, the minimum processing time of 3 time units has to be respected between the end of utilization of B and the beginning of utilization of D.

Train routes b_2 and c_2 share sections s_7 and s_{10} . Since the two trains B and C are traveling in opposite directions, the utilization overlap is jointly computed on s_7 and s_{10} (to avoid a deadlock situation). When train C passes before B on these sections, the utilization overlap is, respectively: 9 (33 - 24). Instead, when train B passes before C no utilization overlap is recorded on s_7 since B finishes using it at 26 and C should start using it at 27. Regarding s_{10} , the cost calculation takes into account the turnaround with D according to Pascariu et al. [22]: s_{10} cannot be occupied by C before D finishes using it. Therefore, the time that C potentially has to wait if B passes first is given by the difference between the end of utilization of s_{10} by D (37) and the beginning of use by C (20), which results to be 17. For each of the two possible train orderings, the maximum utilization overlap is 9 (C before B) and 17 (B before C). In Figure A.1a we illustrate the utilization diagram of the two potential decisions. The minimum between these two values gives the fixed component of the potential delay $w_{b_2c_2}^f = 9$. In this case, train B is the potential waiting train.

Regarding train routes b_2 and d_3 , these are the routes of the trains involved in the turnaround on section s_{10} . The second train in the rolling stock re-utilization, D, can only start using this section after B has finished plus the processing time (3 time units). Since this process time is respected, no utilization overlap is recorded: train D starts utilizing s_{10} at 32, while B ends using s_{10} at 29. As a result, $w_{b_2d_3}^f = 0$ and train D passes necessarily after B.

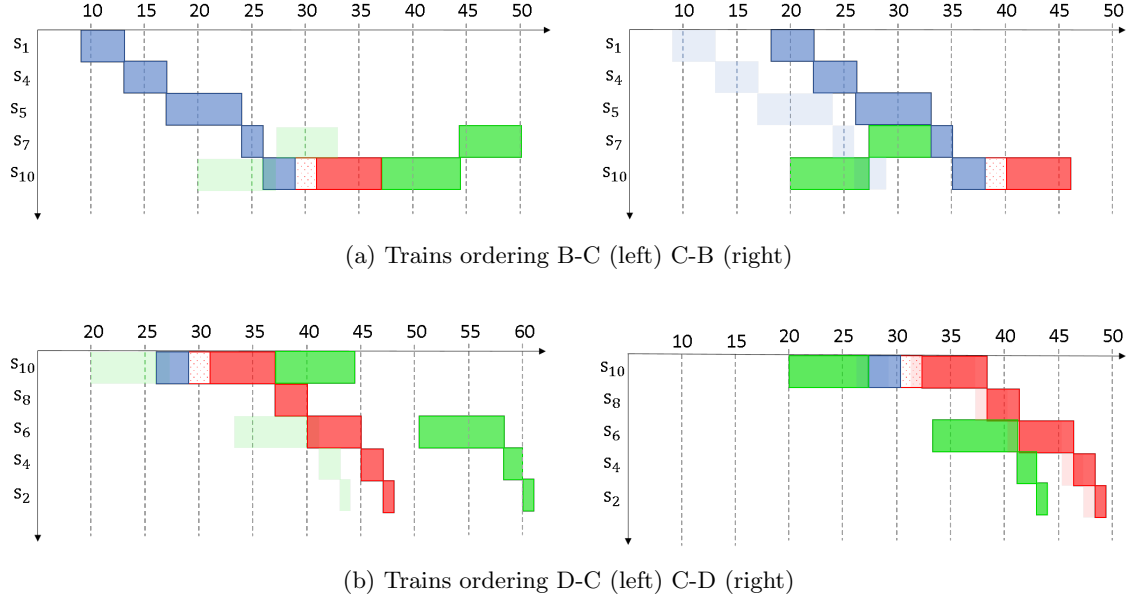


Figure A.1: Illustrative example

Finally, considering the train routes c_2 and d_3 , their cost is calculated in a mirror way with respect to b_2 and c_2 . Train routes b_2 and d_3 share the sections s_{10}, s_6, s_4 and s_2 . When train D passes before C on these sections, the utilization overlap is, respectively: 17 (37 - 20), 12 (45 - 33), 5 (46 - 41) and 5 (48-43). Instead, when we consider the C train passing first, we take into account the rolling stock re-utilization connection on s_{10} . That is, in this section, C must pass before B, the first train in the rolling stock connection. So for the utilization overlap, we consider the difference between the end of utilization of C and the beginning of utilization of B, since D would be delayed consequently. As a result, when train C passes before D on s_{10}, s_6, s_4 and s_2 , the utilization overlap is, respectively: 1 (27 - 26), 1 (41 - 40), -2 (43 - 45) and -2 (44 - 46). For each of the two possible train orderings, the maximum utilization overlap is 17 (D before C) and 1 (C before D). In Figure A.1b we illustrate the utilization diagram of these two potential decisions. The minimum between these two values gives the fixed component of the potential delay $w_{c_2 d_3}^f = 1$. In this case, train D is the potential waiting train.