Omega 137 (2025) 103341

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

# On the fragility of a train timetable

M.L. Tessitore [a], G. Sartor [b],*, M. Samà [a], C. Mannino [b], D. Pacciarelli [a]

[a] Roma Tre University, Via della Vasca Navale 79, 00146 Roma, Italy
[b] SINTEF Digital, Forskningsveien 1, 0373 Oslo, Norway

ARTICLE INFO

*Keywords:*
Robustness
Train timetabling
Mixed-integer programming
Train scheduling

ABSTRACT

In dense railway traffic, even the small delay of a single train may propagate through the network, causing a sequence of knock-on delays and significant deviations from the timetable. To mitigate the propagation of delays, railway infrastructure managers can generally employ two strategies: (i) in real time, train dispatchers can limit delay propagation by rescheduling trains, i.e., by adjusting the timetable in response to disturbances; (ii) in the timetable design phase, train planners can try to build delay-resilient timetables, which requires a complicated and lengthy iterative process. The ability of a timetable to "absorb" delays whenever they occur is known as robustness. While there is no unanimous consensus on a measure to quantify such robustness, it is normally a global measure, and therefore it is unable to highlight whether specific train paths or regions of the railway network are less robust than others. Moreover, only a few academic works incorporate the possibility of mitigating delays in real time through dispatching when evaluating robustness measures. These shortcomings motivate the present work. In this paper, we introduce the concept of fragility as a new practical tool to analyze a given timetable in order to identify the specific sections where a primary delay is most likely to generate knock-on delays, factoring in optimal future dispatching decisions. We also discuss the relationship between the fragility concept and the so-called recovery cost. We present computational results on real-life scenarios from a busy railway line in Norway and discuss several potential uses of fragility to improve decisions at different levels of the railway planning process, including dispatching, timetable design, and network design.

## 1. Introduction

Train timetables are the backbone of rail transport and influence almost all the activities carried out by a railway company, such as personnel scheduling, rolling stock circulation, train and track maintenance. Designing a timetable is thus a very complex planning process that starts months before operations, where many different timetables are compared and evaluated in terms of their ability to satisfy the expected transport demand, their impact on company resources (e.g., personnel, rolling stock, or track capacity) and on the possibility to satisfy other needs (e.g., to ensure that sufficient margins are left in the timetable to perform track maintenance, or to insert additional trains to accommodate peaks of demand during special events).

A critical aspect of the designing process is the evaluation of the timetable's resilience to unexpected events occurring during operations. Planning experts can analyze several perturbed scenarios, assess their potential impact, and suggest changes at different decision-making levels, from tactical-level timetable adjustments to strategic-level infrastructure upgrades. Perturbation can be roughly divided into two categories: disturbances and disruptions. Disturbances are defined as small perturbations that can influence the departure and arrival times of multiple trains in the network. Disruptions, on the other hand, are major breakdowns, such as the partial or full blockage of a track section, leading to a decrease in network capacity. Both disturbances and disruptions may cause an initial delay to a specific train, often called *primary delay*, which may propagate to other trains interacting with it, causing additional *secondary delays* in a snowball effect. However, when a primary delay occurs, recovery actions can be carried out in real time by dispatchers to limit the propagation of secondary delays. Typically, the effect of disturbances can be minimized by relatively small control actions, e.g., reordering or rerouting of trains, while disruption management requires more drastic actions, including trains cancellation, short-turning, rolling stock re-organization, and so on. However, disruptions are more rare when compared to disturbances, which are instead more common and may even occur several times a day [1]. In this study, we focus on disturbance management, because disruptions require a completely different set of considerations and their occurrence is typically ignored while making a timetable.

The literature on disturbance management can be divided in two main approaches and research directions. *Corrective* approaches limit

* Corresponding author.
*E-mail address:* giorgio.sartor@sintef.no (G. Sartor).

https://doi.org/10.1016/j.omega.2025.103341
Received 24 April 2024; Accepted 14 April 2025
Available online 29 April 2025

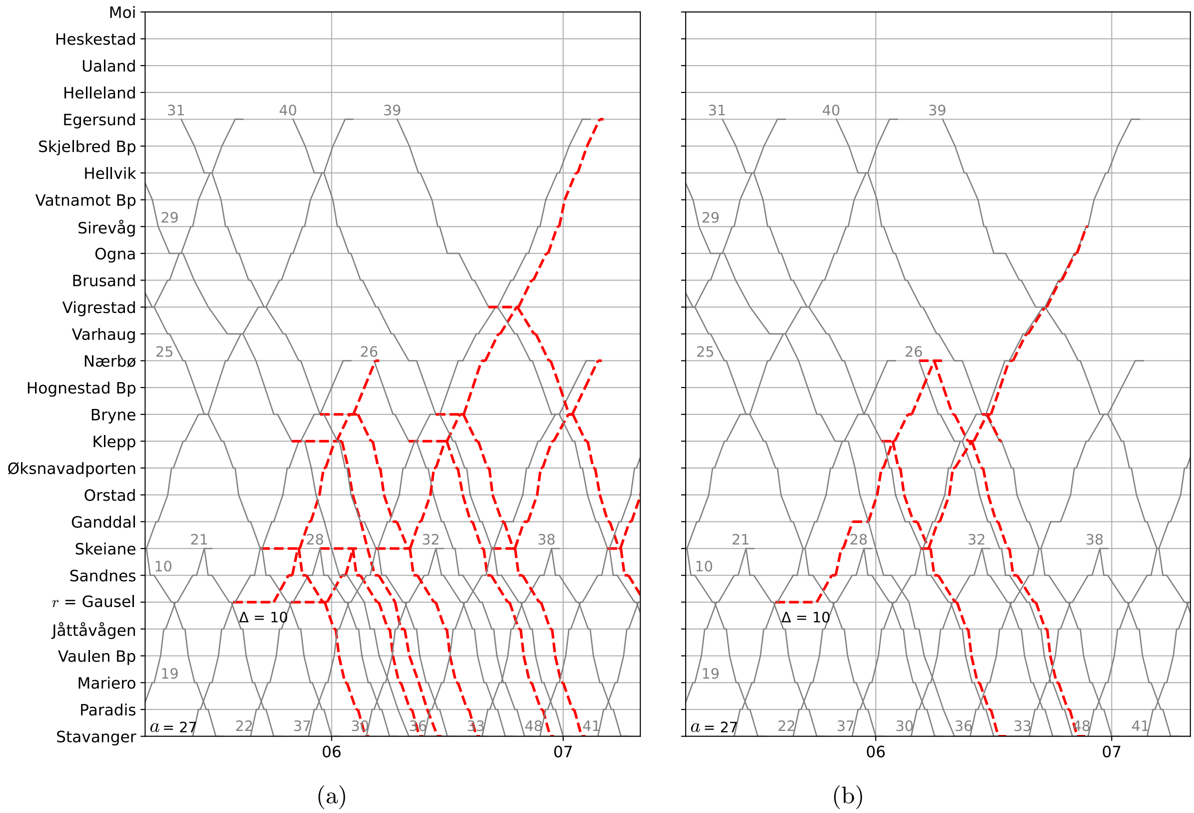



**Fig. 1.** Disposition timetable (a) without re-sequencing and (b) with optimal re-sequencing when the departure of train $a = 27$ from station $r =$ Gausel is delayed by $\Delta = 10$ min.

the delay propagation during operations by recalculating the schedules and/or routes of the circulating trains in real time when they deviate from the original timetable [2–7]. The updated timetable is known as *disposition timetable* [8]. The cost associated with the disposition timetable is a function of its deviation from the original one, and typically increases along with the delays. The second approach is *preventive*, and consists in the design of a *robust* timetable that is able to absorb potential knock-on delays should any disturbance occur. It is normal practice to try to reduce the risk of secondary delays by adding suitable *time supplements* to the train nominal running times, therefore having timetables in which trains are scheduled to run slightly slower than they could. Those time supplements can help absorbing small disturbances and reduce knock-on delays, simply by asking trains to run a bit faster. The ability of a timetable to mitigate secondary delays can be measured by evaluating possible disturbed scenarios. These assessments can then be used in either stochastic or robust optimization methods to build timetables that are resilient to such disturbances. The impact of disturbances is frequently measured via a proxy function that does not consider possible repairing dispatching decisions [9–12]. On the other hand, in the railway practice, both preventive and corrective actions are implemented, especially when dealing with dense timetables where time margins are limited and preventive actions alone are not sufficient to absorb delays and restore punctuality. While considering both preventive and corrective actions is quite uncommon in the literature, a recent work proposes a two-stage stochastic optimization model to deal with disruptive events. The first stage assigns backup rolling stock to storage lines (a preventive action), while the second stage considers rescheduling (corrective) actions [13].

The small example in Fig. 1, from the Norwegian line described in Section 5, shows the importance of corrective actions. In this figure, gray lines represent the nominal timetable for 20 trains circulating between Egersund and Stavanger from 05:00 to 07:20, while dashed red lines highlight the differences between the disposition timetable and the original timetable. In this example, the departure of train $a = 27$ from Gausel station (marked with $r$ in Fig. 1(a)) is delayed by $\Delta = 10$ min, requiring corrective measures. Specifically, Fig. 1(a) shows the disposition timetable when only retiming actions are taken, while the train sequencing prescribed by the original timetable remains unchanged. As shown in the figure, the primary delay $\Delta$ propagates to ten other trains and its effect lasts for more than two hours. The red lines in Fig. 1(b) show instead the disposition timetable when also re-sequencing actions are considered. Here, optimal re-sequencing decisions allow to recover the nominal timetable before 07:00, with a more limited delay propagation (for example, by modifying the timetable to allow trains 27 and 29 to meet at Klepp rather than Gandall, and allowing trains 25 and 27 to cross between Gausel and Sandnes instead of at Skeiane).

The benefit of combining preventive and corrective actions is studied in a stream of papers that considers timetabling design as a two-stage stochastic optimization problem. In their pioneering paper, Liebchen et al. [8] introduce the concept of *recoverable robustness*. Given an optimization problem, a nominal solution for this problem is considered recoverable robust if there exists a recovery algorithm that is able to adjust such a solution for any disturbed realization of the input. In train timetabling, the nominal solution is the planned timetable $\pi$, a realization of the actual traffic scenario $s$ may include primary delays, and the "repairing" algorithm is a train rescheduling algorithm which produces a disposition timetable $\pi'(s)$ for scenario $s$. The *recovery cost* associated with the disposition timetable $\pi'(s)$, produced by the rescheduling algorithm, is then a function $f(z_s)$ of the *deviation* $z_s = d_s(\pi'(s), \pi)$ of timetable $\pi'(s)$ from the nominal timetable $\pi$. The authors propose this approach to solve the *delay management* problem, where main decisions regarding the cancellation of train connections are optimized from the passengers' point of view. This approach sparked a research stream dedicated to recoverable robustness in delay management [14–16].

While the original definition of recoverable robustness makes use of a generic recovery algorithm, Cacchiani et al. [17] propose to employ an exact algorithm returning an optimal solution, therefore making the measure independent of the specific recovery algorithm. In particular, for an input solution and a given scenario, they let the recovery cost be the minimum $f(z_s)$ over all feasible disposition timetables for the given scenario. The authors propose a model that applies the recoverable robustness concept to solve a practical rolling stock rescheduling problem.

Following the recovery robustness approach introduced in Cacchiani et al. [17], in this paper we move a step forward by showing that the recovery cost can be effectively computed on a practical timetable design problem. The main contribution of this paper is the introduction of the new concept of *fragility* as an insightful tool to analyze the resilience of a nominal timetable at a more detailed level, i.e., for measuring its localized ability to recover from primary delays with optimal corrective actions. In particular, given a suitable cost function and a timetable, the fragility defines a special class of scenarios associated with each train–section pair and uses them to identify the pairs where a primary delay is more likely to generate large knock-on delays.

Thus, in contrast to the standard recovery cost [15,17], which is a global measure associated with the entire timetable, fragility is a local measure that is associated with each train–resource pair. It is precisely this locality that makes the fragility such a useful tool for practitioners, as we will see in later sections. The standard recovery cost can then be easily obtained by aggregating the fragility of all train–resource pairs.

Our concept of fragility is also somewhat related to the concept of critical points introduced in [18]. We both try to identify the time-sensitive dependencies between trains at different locations in the network from which a primary delay may propagate. However, [18] only consider the amount of (static) time supplements available to these trains at those locations, without considering real-time corrective actions. This makes the approximation rather unrealistic, because the corrective actions implemented by dispatchers can drastically reduce the delay propagation. In contrast, we consider the actual disposition timetable computed by a (optimal) rescheduling algorithm.

In this paper, we also show that, by computing and analyzing the fragility of a timetable, railway practitioners can learn and extrapolate valuable information that can be used to improve railway management at different decision levels:

- in timetable planning, this information can be used to analyze and later design new robust timetables, for example by adding more time supplements to the most fragile schedules;
- in maintenance planning, the fragility can help to highlight those trains and sections where a reduction of the probability of a disturbance would be the most beneficial, thus suggesting where to improve maintenance or crew scheduling;
- at a strategic level, the fragility can help to direct investments in infrastructure upgrades for improving the train service of a given line, for example by evaluating the fragility of a timetable before and after a station or track upgrade.

This overcomes the major limitation of other existing robustness measures, which are unable to provide specific suggestions or insights regarding where and why certain trains or certain sections of a network may cause large knock-on delays. Moreover, the disposition timetables computed for each delay scenario can be used to preemptively study optimal rescheduling decisions, should any of those scenarios occur.

The rest of the paper is organized as follows. Section 2 introduces the concept of fragility. Section 3 presents a MILP model for the fragility, and Section 4 develops a solution algorithm for efficiently solving this model. Experimental results are analyzed in Section 5, where we show an example of the potential use of the fragility in a practical case, namely the dense timetable of the Jæren line, a railway line located in Southern Norway. Finally, Sections 6 and 7 discuss potential applications of the concept of fragility in different railway planning processes and possible future developments, respectively.

## 2. Definition of fragility

When a train runs through a network, it traverses an ordered sequence of network resources, known as the *route* of the train. The routes of distinct trains may contain conflicting network resources, and the trains need to be suitably sequenced and scheduled on such resources. A nominal timetable $\pi$ is a feasible schedule for all trains running in the railway network, i.e. with no conflicts in the use of resources. Let $A$ be the set of trains running in the network and let $R$ represent the set of network resources. Each train $a \in A$ runs through a sequence of network resources $R_a$. For each train $a \in A$ and network resource $r \in R_a$ in its route, the nominal timetable has an entry time $\pi^r_a$, which is the time train $a$ is planned to enter resource $r$. During operations, if train $a$ gets delayed (e.g., due to a prolonged stop at a station), the nominal timetable is no longer valid and the schedule of train $a$ (and possibly of other trains interacting with $a$) need to be updated. To this end, train dispatchers can make some decisions (for example, changing the nominal routes of the trains or their relative sequence on the contended resources) that implicitly define a new timetable $\pi'$ called *disposition timetable*. An optimal disposition timetable is one that minimizes some cost function $f(\pi', \pi)$ of the deviation from the nominal one, for instance, the total delay of all trains.

The robustness of a timetable is a measure of its ability to "absorb" potential primary delays or, in other words, to minimize the risk of secondary delays. To measure such ability, we will make use of a family of traffic scenarios $\mathcal{S}$, in which one or more trains are delayed. In order to build a feasible disposition timetable for a scenario $s \in \mathcal{S}$, we need to have at hand a dispatching algorithm $\mathcal{A}$. Let $\pi'(\mathcal{A}, s)$ be the disposition timetable returned by $\mathcal{A}$ when the input scenario is $s$. Evidently, the cost $f(\pi'(s, \mathcal{A}), \pi)$ of $\pi'(s, \mathcal{A})$ depends both on the chosen scenario $s$ and on the ability of algorithm $\mathcal{A}$ to generate good disposition timetables. To make the measure independent of the algorithm, Cacchiani et al. [17] suggest considering optimal disposition timetables $\pi^*(s)$ determined by an exact algorithm that computes optimal dispatching decisions. Accordingly, we let $f(\pi^*(s), \pi) = \min_{\mathcal{A}} f(\pi'(s, \mathcal{A}), \pi)$ be the *recovery cost* of scenario $s$ for the nominal timetable $\pi$. Moreover, if we have at hand a probability distribution for the set $\mathcal{S}$ of delay scenarios, one can compute the recovery cost for $\pi$ as the expected value $\mathbb{E}_s[f(\pi^*(s), \pi)]$ with respect to this given distribution. Roughly speaking, a timetable $\pi$ is recoverable robust whenever it admits a small recovery cost.

So, the recovery cost in Cacchiani et al. [17] is a global measure of the timetable that does not provide any information regarding, e.g., how sensitive is the timetable to local perturbations in specific segments of the railway network, at a particular time of the day, and how difficult it may be to recover from such perturbations. Indeed, having such local information would greatly help railway practitioners analyze the performance of a timetable and plan adjustments.

More specifically, given a timetable $\pi$, we want to identify its *fragile* sections by assessing the effectiveness of the optimal dispatching given sets of scenarios, with the aim of improving the timetable robustness by intervening on these sections. To compute this local information, we define a particular set $\mathcal{S}$ of delay scenarios such that each scenario contains exactly one primary delay (*single-delay scenario*). More precisely, each scenario $s = (a, r, \Delta) \in \mathcal{S}$ is associated with a train $a \in A$ being delayed by $\Delta$ min in resource $r$. We concentrate on single-delay scenarios since the probability of having multiple simultaneous unexpected events resulting in primary delays in the network is much smaller than that of a single one. Also, this allows us to create the graph in Fig. 7, which gives railway practitioners an immediate visual feedback on the localized fragility of a timetable.

**Definition 2.1.** Given a timetable $\pi$, a train $a \in A$, a resource $r \in R_a$ and a set of scenarios $\mathcal{S}_{ar} = \{(a, r, \Delta_i) : i = 1, \ldots, n\}$, with an associated discrete probability distribution $\rho_{ar}$, we define the fragility $F(\pi, \mathcal{S}_{ar})$ of the train–resource pair $(a, r)$ as the expected recovery cost of $\pi$ with respect to the scenarios in $\mathcal{S}_{ar}$, namely:

$$F(\pi, \mathcal{S}_{ar}) = \sum_{i=1}^{n} \rho^i_{ar} \cdot f(\pi^*(a, r, \Delta_i), \pi).$$

In the rest of the paper, we say that a train–resource pair $(a, r)$ is *fragile* if it has a high expected recovery cost $F(\pi, \mathcal{S}_{ar})$. If recoverable robustness describes how well a timetable can recover from potential primary delays, the fragility of $(a, r)$ quantifies the extent to which potential delays in that section will affect the timetable, even when considering the best possible recovery strategy. A high level of fragility denotes a critical train–resource pair, where the occurrence of primary delays will generate substantial knock-on effects.

Finally, observe that a global robustness measure for timetable $\pi$ can be easily obtained as a function of the individual fragility costs. We can start by defining a (global) fragility measure or recovery cost $F(\pi, \mathcal{S})$ of $\pi$. For instance, we can let $F(\pi, \mathcal{S}) = max_{a\in A, r\in R_a}\{F(\pi, \mathcal{S}_{ar})\}$; or, alternatively, $F(\pi, \mathcal{S}) = \sum_{a\in A, r\in R_a}\{F(\pi, \mathcal{S}_{ar})\}$. Then, the robustness $\mathcal{R}(\pi, \mathcal{S})$ of $\pi$ can be defined as the inverse of the recovery cost, namely $\mathcal{R}(\pi, \mathcal{S}) = \frac{1}{F(\pi, \mathcal{S})}$.

## 3. Modeling the dispatching problem

To compute the fragility of a pair $(a, r)$, we need to solve a dispatching problem for all scenarios in set $\mathcal{S}_{ar}$. To highlight the concept of fragility and for ease of explanation, we consider a slightly simplified model for these scenarios, designed for the Jæren line in Norway. We base this model on the one presented by Lamorgese and Mannino [5], which was later validated in practical, real-life settings (see, for example, Bach et al. [2] and Sartor et al. [19]). Since the definition of fragility is independent from the formulation of the dispatching problem, more complex settings can be studied by suitable adaptations of the model presented in this section.

We consider a macroscopic representation of the network (see Pachl and White [20]), in which a station is "collapsed" to an individual network resource capable of accommodating up to a given number of trains at a time. In other words, we neglect the microscopic topology of the station (for example, crossing tracks or platforms), nor do we model the exact movements of trains in stations. This approximation can be easily removed with a more comprehensive model, but it is particularly acceptable in the railway line we are considering, which only contains "simple" stations with up to 5 platforms. Each pair of adjacent stations is connected by at most two tracks. In double-track sections, one track is normally reserved for trains running in one direction, while the other track is reserved for the opposite direction. In single-track sections, trains traveling in opposite directions must alternate. In some cases, a track connecting two stations is long enough that it is considered as two separate track sections. This can be easily modeled by adding a virtual station (sometimes called *blockpoint*) separating the two track sections, so that two trains running in the same direction can be simultaneously in the two different track sections. Those stations are identified with the abbreviation "Bp" at the end of their name. Besides respecting all physical and safety constraints, any (feasible) timetable must ensure that no two trains are on the same track at the same time and that no station contains more trains than its capacity.

We now present a model $\mathcal{P}_s$ to compute the recovery cost $f(\pi^*(\tilde{a}, \tilde{r}, \tilde{\Delta}), \pi)$ of a specific scenario $s = (\tilde{a}, \tilde{r}, \tilde{\Delta})$ for a timetable $\pi$, necessary for the computation of the fragility of a pair $(a, r)$. For ease of reference, a comprehensive summary of the notation used in this model is provided in Table 1. The railway line under consideration has a set $R$ of railway resources, partitioned into stations $R^S$ and tracks $R^T$ connecting the stations. A station $r \in R^S$ has capacity $C^r \in \mathbb{Z}$ representing the number of trains which can be simultaneously in the station. Each track $r \in R^T$ can be considered as a resource with unit capacity. Let $A$ be the set of trains running in the network, each train $a \in A$ runs through an ordered alternating sequence $R_a = (r^1_a, r^2_a, \ldots, r^{q_a}_a) \subseteq R$ of stations and tracks (its route). Variables $p^r_a$ and $\underline{p}^r_a$ represent, respectively, the nominal and minimum time the train spends in resource $r \in R_a$, where the first and last resources are always stations. Note that a primary delay $\Delta$ for train $a$ on resource $r \in R_a$ can be defined as an increase of its processing time to $p^r_a + \Delta$. The nominal schedule of train $a$ is a vector $\pi_a \in \mathbb{R}^{R_a}_+$, where, for $r \in R_a$, $\pi^r_a$ is the time train $a$ enters resource $r$. The nominal timetable is therefore defined as the set of vectors $\pi = \bigcup_{a\in A} \pi_a$. We assume that $\pi$ is conflict free (curiously enough, real nominal timetables are not always completely conflict free). We associate a scheduling variable $t^r_a \in \mathbb{R}$ with each resource occupation, representing the time train $a$ enters resource $r$ in the disposition timetable. We also introduce a new variable $t^{out}_a \in \mathbb{R}$ to represent the time train $a$ leaves the last resource of its route. Additionally, we include a fictitious variable $t^0 \in \mathbb{R}$ representing the reference initial time of the overall timetable. Any other time is considered an offset with respect to $t^0$. In the following, we describe the three groups of model constraints and the considered objective function.

*Free-running constraints.* The first group of constraints models the behavior of each single train as if it was traveling alone in the network.

*Start constraints.* A train $a \in A$ cannot start its route before the nominal start time $\pi^{r^1}_a$:

$$t^{r^1}_a - t^0 \geq \pi^{r^1}_a, \qquad a \in A. \tag{1}$$

*Departure constraints.* Let $A^P$ be the set of passenger trains running in the network, with $A^P \subseteq A$. No passenger train can leave a station (i.e., enter the next resource) before the time prescribed by the nominal timetable:

$$t^{r+1}_a - t^0 \geq \pi^{r+1}_a, \qquad a \in A^P, r \in R^S_a \setminus \{r^{q_a}_a\}. \tag{2}$$

Here, for $r \in R_a$ different from the destination station of $a$, we let $t^{r+1}_a$ be the scheduling variable associated with the resource immediately following $r$ on the route of $a$, i.e., the time $a$ leaves resource $r$, and $\pi^{r+1}_a$ the departure time prescribed in the timetable. Note that if for some train $a$ and station $r \in R^S_a$ such time is not given, we assume $\pi^{r+1}_a = 0$.

*Running time constraints.* The time spent by a train in each resource is at least the minimum running time $\underline{p}^r_a$:

$$t^{r+1}_a - t^r_a \geq \underline{p}^r_a, \qquad a \in A, r \in R_a. \tag{3}$$

When $r$ is the final station on the route of train $a$, i.e., $r = r^{q_a}_a$, we let $t^{r+1}_a = t^{out}_a$.

The system of constraints (1)–(3) is called free-running because it models feasible schedules when trains do not interact with each other.

*Conflict avoiding constraints.* In general, trains do interact with each other, and they compete to access the (scarce) resources of the network. Following Lamorgese and Mannino [5], we handle potential conflicts arising in stations and tracks in a slightly different way.

In single-track sections, trains in both directions will share the unique track connecting adjacent stations. In double-track sections, each track is reserved to a specific direction, and only trains in the same direction use the same track. In our model, each track represents a resource of single unit capacity. Therefore, no two trains can occupy simultaneously the same track.

As for stations, we will not model directly the individual movements of a train within the station. Instead, we will look at each station $r \in R^S$ as a "capacitated" macroscopic resource. Let $A(r) \subset A$ be the set of trains traversing $r$ and let $Q \subseteq A(r)$ be a subset of trains traversing $r$ at the same time (i.e., the intersection of the time intervals spent by each train $a \in Q$ in station $r$ is nonempty). In our constraints, we consider the family $Q(r) \subseteq 2^{A(r)}$ of the minimal infeasible subsets of trains that can traverse station $r$ at the same time. Here minimal means that, for any

**Table 1**
List of sets, parameters and variables used in the proposed model.

| Notation | Type | Description |
|---|---|---|
| $A$ | Set | Set of trains |
| $A^P$ | Set | Set of passenger trains |
| $R$ | Set | Set of network resources |
| $R^S$ | Set | Set of stations |
| $R^T$ | Set | Set of tracks |
| $R_a$ | Set | Set of resources in the route of train $a$ |
| $R_a^S$ | Set | Set of stations in the route of train $a$ |
| $A(r)$ | Set | Set of trains traversing resource $r$ |
| $\mathcal{Q}(r)$ | Set | Minimal infeasible subsets of trains traversing station $r$ |
| $r_a$ | Param | Resource $r$ in the route of train $a$ |
| $r_a^{q_a}$ | Param | Last resource in the route of train $a$ |
| $\pi_a^r$ | Param | Aimed arrival time of train $a$ at resource $r$ |
| $\underline{p}_a^r$ | Param | Minimum running time of train $a$ on resource $r$ |
| $p_a^r$ | Param | Aimed running time of train $a$ on resource $r$ |
| $h_a^r$ | Param | Required safety margin for train $a$ in station $r$ |
| $d_s^U$ | Param | Unrecoverable delay of train $\tilde{a}$ in scenario $s=(\tilde{a},\tilde{r},\tilde{\Delta})$ |
| $C^r$ | Param | Capacity of station $r$ |
| $\Delta$ | Param | Primary delay |
| $t^0$ | Var | Timetable reference entry time |
| $t_a^r$ | Var | Arrival time of train $a$ at resource $r$ |
| $t_a^{out}$ | Var | Exit time of train $a$ from the infrastructure |
| $x_{ab}^r$ | Var | Binary variable that is 1 if trains $a$ and $b$ meet in station $r$ |
| $y_{ab}^r$ $(y_{ba}^r)$ | Var | Binary variable that is 1 if train $a$ $(b)$ enters resource $r$ before train $b$ $(a)$ |
| $w_{ab}^r$ $(w_{ba}^r)$ | Var | Binary variable that is 1 if trains $a$ and $b$ meet in $r$ and $a$ $(b)$ enters first |
| $d_a^{out}$ | Var | Delay of train $a$ at its destination station |

$Q \in \mathcal{Q}(r)$ and any $a \in Q$, the set $Q \setminus \{a\}$ can be accommodated in $r$. We remark that we are neglecting the more general case in which a certain subset of trains $Q$ may be accommodated in a station $r$ only for certain specific arrival or exit times. This is very unlikely to happen in the railway line under consideration. A way to handle this in a macroscopic decomposition approach has been recently developed in Leutwiler and Corman [21].

*Track conflict constraints.* Let $r \in R^T$ be a track, and $a, b \in A(r)$ be two different trains traversing $r$. We need to ensure that either $a$ runs through and exits $r$ before $b$ enters $r$, or viceversa. To this end, we introduce a binary variable $y_{ab}^r$ for every ordered pair of distinct trains $a, b \in A(r)$, which is 1 if and only if $a$ precedes $b$ in $r$. This is expressed by the following constraints:

$$t_b^r - t_a^{r+1} \geq M(y_{ab}^r - 1), \qquad a, b \in A(r), a \neq b, r \in R^T, \tag{4}$$

where $M$ is a large positive constant (the notorious *big-M*). Indeed, the constraints are active when $a$ precedes $b$ (and $y_{ab}^r = 1$), and become redundant in the other case. Moreover, it must be that only one train precedes the other:

$$y_{ab}^r + y_{ba}^r = 1, \qquad a, b \in A(r), a \neq b, r \in R^T. \tag{5}$$

*Station capacity constraints.* The main difference with the track conflict constraints is that a station $r \in R^S$ may (in general) accommodate more than one train at a time. To model this, we introduce a binary variable $x_{ab}^r$ for each (unordered) pair of distinct trains $a, b \in A(r)$, which is 1 if and only if $a$ and $b$ meet in $r$ (i.e., the intersection of the time intervals spent by trains $a$ and $b$ in station $r$ is nonempty). The next constraints model the fact that either $(i)$ trains $a$ and $b$ meet in $r$, or $(ii)$ train $a$ exits $r$ before $b$ enters $r$, or $(iii)$ train $b$ exits $r$ before $a$ enters $r$:

$$y_{ab}^r + y_{ba}^r + x_{ab}^r = 1, \qquad a, b \in A(r), a \neq b, r \in R^S. \tag{6}$$

Now, when $x_{ab}^r = 1$, trains $a$ and $b$ meet in $r$, which implies $a$ enters the station before $b$ exits and viceversa. This is ensured by the next pair of constraints:

$$\begin{cases} t_b^{r+1} - t_a^r \geq M(x_{ab}^r - 1) \\ t_a^{r+1} - t_b^r \geq M(x_{ab}^r - 1) \end{cases} \qquad a, b \in A(r), a \neq b, r \in R^S. \tag{7}$$

Constraints (8) are analogous to (4). They imply that if a pair of trains does not meet in station $r$, the one winning the conflict must exit the station before the other enters.

$$t_b^r - t_a^{r+1} \geq M(y_{ab}^r - 1) \qquad a, b \in A(r), a \neq b, r \in R^S. \tag{8}$$

Constraints (9) ensure that, for any set $Q$ of trains in $\mathcal{Q}(r)$, at least one pair of trains does not meet in $r$ (recall that the total number of train pairs in $Q$ is $\binom{|Q|}{2}$):

$$\sum_{\{a,b\} \subseteq Q} x_{ab}^r \leq \binom{|Q|}{2} - 1, \qquad Q \in \mathcal{Q}(r), r \in R^S. \tag{9}$$

*Safety margin constraints.* For safety reasons, when two trains go through a common station, the second train is required to enter the station only some time after the first train fully arrived at that station. This time interval is usually called safety margin, and its value $h_a^r$ depends both on train $a$ and station $r$. When two trains running in opposite directions meet in a station that can accommodate more than one train at a time, the second train must enter the station only some time after the first train entered that station. However, when two trains run in the same direction, or move in opposite directions but meet in a single-capacity station, a safety margin is still guaranteed by the timetable.

We introduce a binary variable $w_{ab}^r$ for every $(a, b) \in A(r) \times A(r), a \neq b, r \in R^S$ with $C^r > 1$, which is 1 if and only if trains $a, b$ meet in $r$ and $a$ enters first. In particular, assume that trains $a, b$ are crossing trains meeting in a station $r$ that is not of single capacity. If train $a$ $(b)$ enters first, then train $b$ $(a)$ can enter $r$ at least $h_a^r$ $(h_b^r)$ time units after the arrival of $a$ $(b)$ at $r$:

$$\begin{cases} w_{ab}^r + w_{ba}^r - x_{ab}^r = 0 \\ t_b^r - t_a^r \geq M(w_{ab}^r - 1) + h_a^r \\ t_a^r - t_b^r \geq M(w_{ba}^r - 1) + h_b^r \end{cases} \qquad a, b \in A(r), a \neq b, r \in R^S, C^r > 1. \tag{10}$$

*Scenario-specific constraints.* We are finally ready to introduce variables and constraints specifically related to scenario $(\tilde{a}, \tilde{r}, \tilde{\Delta})$ in which train $\tilde{a}$ is delayed by $\tilde{\Delta}$ time units in resource $\tilde{r} \in R_{\tilde{a}}$.

*Extra running-time constraints.* A delay $\tilde{\Delta} > 0$ is added to the original running time of $\tilde{a}$ on resource $\tilde{r}$:

$$t_{\tilde{a}}^{\tilde{r}+1} - t_{\tilde{a}}^{\tilde{r}} \geq p_{\tilde{a}}^{\tilde{r}} + \tilde{\Delta}, \quad \tilde{a} \in A, \tilde{r} \in R_{\tilde{a}}. \tag{11}$$

*Non-anticipative recovery algorithm constraints.* As previously discussed, recovery algorithms are non-anticipative, so we cannot take decisions which modify the timetable before the primary delay happens. Then, we have that:

$$t_a^r = \pi_a^r, \qquad a \in A, r \in R_a : \pi_a^r \le \pi_{\tilde{a}}^{\tilde{r}+1}. \tag{12}$$

In other words, all schedules follow the nominal timetable until the delay occurs.

*Objective function.* The objective function to be optimized in this model has to be the same function $f$ dispatchers take into consideration in real-time. Given the lack of a generally recognized objective used in real-time, we settle on one among the most popularly used in the literature, the minimization of the sum of consecutive train delays at the last station of their routes, also known as *total delay* [12,22–25]. Still, other cost functions can be considered as well, requiring only to update the objective function and use additional constraints, if necessary.

*Delay constraints*. Let $\pi_a^{out}$ be the nominal departure time of train $a \in A$ from its destination station. We introduce a non-negative variable $d_a^{out} \in \mathbb{R}_+$ to represent the delay of each train $a \in A$ at its destination station:

$$t_a^{out} - d_a^{out} \le \pi_a^{out}, \qquad a \in A. \tag{13}$$

*Fragility of scenario $s$.* When a train is affected by a primary delay $\Delta$ at some point in its route, it could, in principle, regain partial or full punctuality at destination by (if possible) running faster or exploiting time supplements. If we neglect the presence of other trains, we can easily compute the maximum time delay that the train can recover. This may still not be sufficient to regain full punctuality at destination, and the residual delay is what we define to be *unrecoverable*, because there is nothing we can do to reduce it further. So, this unrecoverable delay is independent of any dispatching decision, and we want to factor it out when assessing the quality of the plan. This motivates the following definition. Given scenario $s = (\tilde{a}, \tilde{r}, \tilde{\Delta}) \in \mathcal{S}$, we introduce a parameter $d_s^U$ corresponding to the minimum *unrecoverable* delay of scenario $s$, i.e., the delay at destination of train $\tilde{a}$ (when delayed by $\tilde{\Delta}$ on $\tilde{r}$) that can never be recovered regardless of dispatching decisions, even when $\tilde{a}$ is considered as free-running in the network. The recovery cost of scenario $s$ is computed minimizing the sum of all train delays at destination minus its minimum unrecoverable delay:

$$f(\pi^*(\tilde{a}, \tilde{r}, \tilde{\Delta}), \pi) = \left( \sum_{a \in A} d_a^{out} \right) - d_s^U. \tag{14}$$

Solving the MILP model $\mathcal{P}_s$ for computing the deviation cost associated to scenario $s = (\tilde{a}, \tilde{r}, \tilde{\Delta}) \in \mathcal{S}$ corresponds to minimize $f(\pi^*(\tilde{a}, \tilde{r}, \tilde{\Delta}), \pi)$, subject to constraints (1)–(13).

Intuitively, one would expect the recovery cost to increase monotonically as the primary delay $\Delta$ increases. In fact, this would be the case if we ignored the minimum unrecoverable delay $d_s^U$ in the objective function. However, due to the chosen dispatching options, it may happen that the value of the objective function paradoxically decreases as the primary delay grows. In the next section, we will explain this paradox and discuss why we decided to include the minimum unrecoverable delay in the objective function.

### 3.1. Fragility paradox

Let us consider a train $a = 1$ running on a small railway line of four stations, i.e, from $r^1$ to $r^4$. Figs. 2 and 3 show a schematic explanation of the core mechanism of model $\mathcal{P}_s$ for computing the recovery cost $f(\pi^*(1, r, 5), \pi)$ for three scenarios, with $r = \{r^1, r^2, r^3\}$. In these figures, the stations are depicted on the $y$-axis, the $x$-axis represents the time interval [9:00–10:00], the light-colored lines represent the nominal timetable of train 1, and the dark lines its actual behavior (i.e., the disposition timetable). Given the non-anticipatory nature of train dispatching, train operations preceding the departure from the delayed station are fixed in time.

In Fig. 2, we consider three different scenarios $(1, r, 5)$ where train 1 is delayed at different stations along its route, i.e., $r^1$, $r^2$ and $r^3$. In Fig. 2(a), the train is delayed at the beginning of its route and it is able to completely recover the primary delay when leaving the last station of its route, i.e., $d_1^{out} = 0$. However, when delaying the departure from subsequent stations (Figs. 2(b) and 2(c)), the train leaves its destination station with $d_1^{out} = 1$ and $d_1^{out} = 3$, respectively. Since the train is traveling undisturbed, these delays turn out to be exactly the minimum unrecoverable delays for the given scenarios.

Fig. 3 presents the fragility values corresponding to the three scenarios in Fig. 2, when unrecoverable delays are (a) included or (b) excluded from the computation. A color ranging from yellow to black is associated with each scenario, representing its fragility value, i.e., the darker the color, the more fragile that part of the schedule. In order to understand why it is important to include the minimum unrecoverable delay in the recovery cost, let us first consider a recovery cost that is simply equal to the sum of trains delays at their final station. For simplicity, let assume that a track following a station in the route of a train inherits the fragility value associated with that station. From Fig. 3(a), we can see that a resource at the end of the train schedule will always be more fragile than a resource at the beginning of the same schedule. For example, when delaying the departure of the train from station $r^3$, the train gains a final delay $d_1^{out} = 3$ min, hence, both station $r^3$ and the track following $r^3$ are represented in black. In this case, the train does not have enough time to recover the primary delay, since most of its schedule has already been run. Of course, the situation is very different if we delay the train departure from station $r^1$, because the train would be able to catch up by running faster to the end of the schedule, i.e., $d_1^{out} = 0$, hence the yellow color assigned to station $r^1$ and the following track.

However, would it make sense to say that the schedule of a train is more fragile closer to its destination? These results are obvious and do not provide useful information on where the actual weaknesses of a timetable are. To solve this issue, we include the unrecoverable delay $d_s^U$ in the objective function, which results in Fig. 3(b). For example, when delaying the departure of the train from station $r^3$, the train gains a final delay $d_1^{out} = 3$ min, that the train is not able to recover. Since there are no other trains in the network, the total delay of the train at its final destination always corresponds to the unrecoverable delay of the scenario, i.e., $d_s^U = d_1^{out}$, and all scenarios have a recovery cost equal to 0.

Fig. 4 shows an illustrative example of the recovery cost obtained when train 1, depicted in blue, is delayed from station $r^2$ by $\Delta = 5$ min (Fig. 4(b)) and $\Delta = 12$ min (Fig. 4(a)), respectively. On the left hand side of Fig. 4, we show what would happen to train 1 if it was free-running in the network: with $\Delta = 5$ min, the train would be able to recover its primary delay, leaving station $r^4$ on time, i.e., $d_s^U = 0$; differently, when $\Delta = 12$ min, the train would be unable to accomplish this goal, leaving $r^4$ with delay $d_1^{out} = d_s^U = 7$ min. On the right hand side of Fig. 4, we show what happens when train 1 uses the line alongside a second train, depicted in green. In the optimal disposition timetable, when $\Delta = 5$ min, the two trains exit their final station with a delay of 5 and 4 min, respectively. However, when $\Delta = 12$ min, train 2 leaves $r^4$ on time, while train 1 has a final delay of 14 min. In this case, the total delay at destinations becomes $d_1^{out} + d_2^{out} = 9$ min when $\Delta = 5$ min, and $d_1^{out} + 0 = 14$ min when $\Delta = 12$ min. As a result, the total delay increases with the size of the primary delay $\Delta$.

However, if we evaluate the unrecoverable delay $d_s^U$ in the objective function, the objective value does not necessarily increase with the size of primary delay $\Delta$. This means that we may have a scenario $(\tilde{a}, \tilde{r}, \Delta)$ that is fragile when considering a small delay, but less fragile with a higher

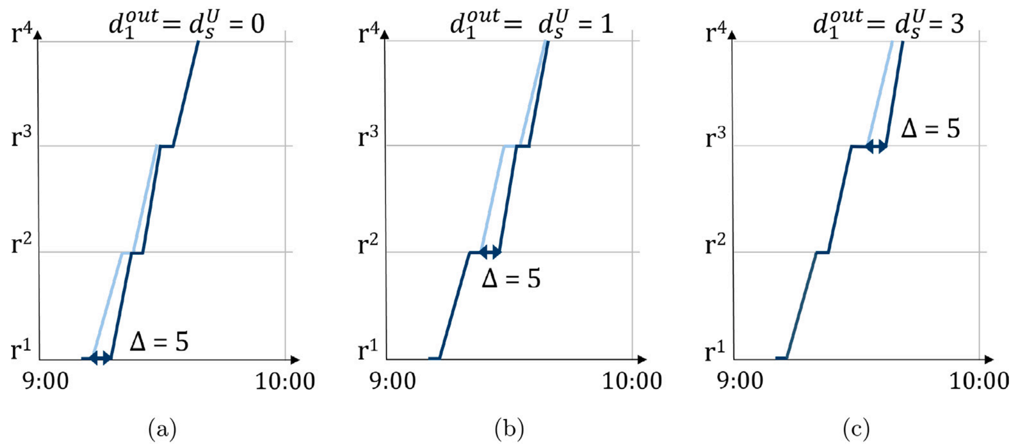



**Fig. 2.** Optimal dispatching decisions associated with three delay scenarios $(\mathbf{1}, \boldsymbol{r}, \mathbf{5})$, where $\boldsymbol{r}$ corresponds, respectively, to station (a) $\boldsymbol{r}^1$, (b) $\boldsymbol{r}^2$ and (c) $\boldsymbol{r}^3$ in the route of train **1**.

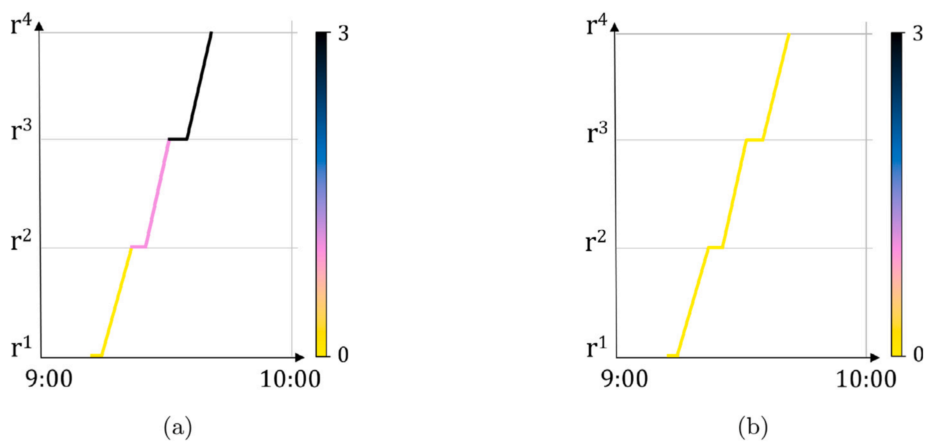



**Fig. 3.** Fragility values corresponding to delay scenarios in Fig. 2 and objective values (a) **min** $\sum_{a \in A} d_a^{out}$ and (b) **min** $\sum_{a \in A} d_a^{out} - d_s^U$.

one, e.g., in Fig. 4, the objective value is $d_1^{out} + d_2^{out} - d_s^U = 9$ min when $\Delta = 5$ min, but $d_1^{out} + d_2^{out} - d_s^U = 7$ min when $\Delta = 12$ min. This paradox is a direct consequence of the chosen dispatching decisions (e.g., in Fig. 4 the two trains meet in station $r^2$ instead of $r^3$ as originally scheduled) as well as of the given definition of fragility, which requires the possibility of improving the timetable.

## 4. Solution algorithm

In this section, we present the algorithm developed to compute the fragility of a train–resource pair $(a, r)$.

As previously said, to compute the fragility of a pair $(a, r)$, we should solve a dispatching problem $\mathcal{P}_s$ for each scenario $s = (a, r, \Delta) \in \mathcal{S}$. A key aspect to speed up the solution process becomes identifying the minimal subset of variables and constraints necessary to find the (feasible) optimal solution for the whole problem. This can be achieved by efficiently exploiting the structure of the MILP model defined in Section 3 and by dynamically generating all the necessary constraints and related variables, through a delayed row generation scheme, similar to the one first described in Lamorgese and Mannino [5]. In particular, we apply the row generation scheme on conflict avoiding constraints (4) and (9), as shown in Algorithm 1.

Let us focus on a specific train–resource pair $(a, r)$ and the set of scenario $\mathcal{S}_{ar}$ it involves. From these scenarios, we derive the ordered set of delays $\{\Delta_1, \ldots, \Delta_n\}$, with $\Delta_{j-1} < \Delta_j$ and $j = 2, \ldots, n$. We initially build the MILP model $\mathcal{P}_{(a,r)}$, including all free running constraints (1)–(3), in which each train behaves as if traveling alone in the network, the delay constraints (13), and the objective function (14) (line 2 of Algorithm 1).

Instead of directly solving $\mathcal{P}_{(a,r,\Delta_1)}$ for the specific scenario $s = (a, r, \Delta_1)$, we first build model $\mathcal{P}^0_{(a,r,\Delta_1)}$ from $\mathcal{P}_{(a,r)}$, by adding the corresponding scenario-specific constraints (11)–(12) (line 5). It is important to note that, at this point, none of the variables $y^r_{ab}$, $x^r_{ab}$ and $w^r_{ab}$ have been included into the current model $\mathcal{P}^0_{(a,r,\Delta_1)}$.

We first solve $\mathcal{P}^0_{(a,r,\Delta_1)}$ with its limited subset of variables and constraints and check if its feasible, optimal solution $\pi^0$ is also feasible for the overall $\mathcal{P}_{(a,r,\Delta_1)}$. Namely, we check if any conflict avoiding constraints (4)–(10) not included in problem $\mathcal{P}^0_{(a,r,\Delta_1)}$ is violated by the current solution $\pi^0$. If so and, e.g., two trains occupy simultaneously the same track or a single-capacity station, we build a larger $\mathcal{P}^1_{(a,r,\Delta_1)}$ from $\mathcal{P}^0_{(a,r,\Delta_1)}$ in which we add all violated constraints and the related $y^r_{ab}$, $x^r_{ab}$ and $w^r_{ab}$ variables.

When we introduce these new variables, we also include in $\mathcal{P}^1_{(a,r,\Delta_1)}$ all constraints that affect their values, even if those constraints were not originally violated. Indeed, let us remind that: $(i)$ variables $y^r_{ab}$ appear only in track and station conflict constraints (4)–(6) and (8), while variables $x^r_{ab}$ appear only in station conflict constraints (6)–(7) and safety margin constraints (9)–(10); $(ii)$ constraints (5)–(8) and (10) are only relevant in conjunction with, respectively, constraints (4) and (9), which are the most relevant in the conflict avoidance group. Therefore, we can delay the generation of constraints (5)–(8) and (10), and variables $y^r_{ab}$ and $x^r_{ab}$, until constraints (4) and (9) need to be generated.

We solve the new model $\mathcal{P}^1_{(a,r,\Delta_1)}$ and check the feasibility of its solution $\pi^1$ against the overall $\mathcal{P}_{(a,r,\Delta_1)}$. We keep iterating, as shown in lines 9-23, considering a sequence of increasingly large MILPs

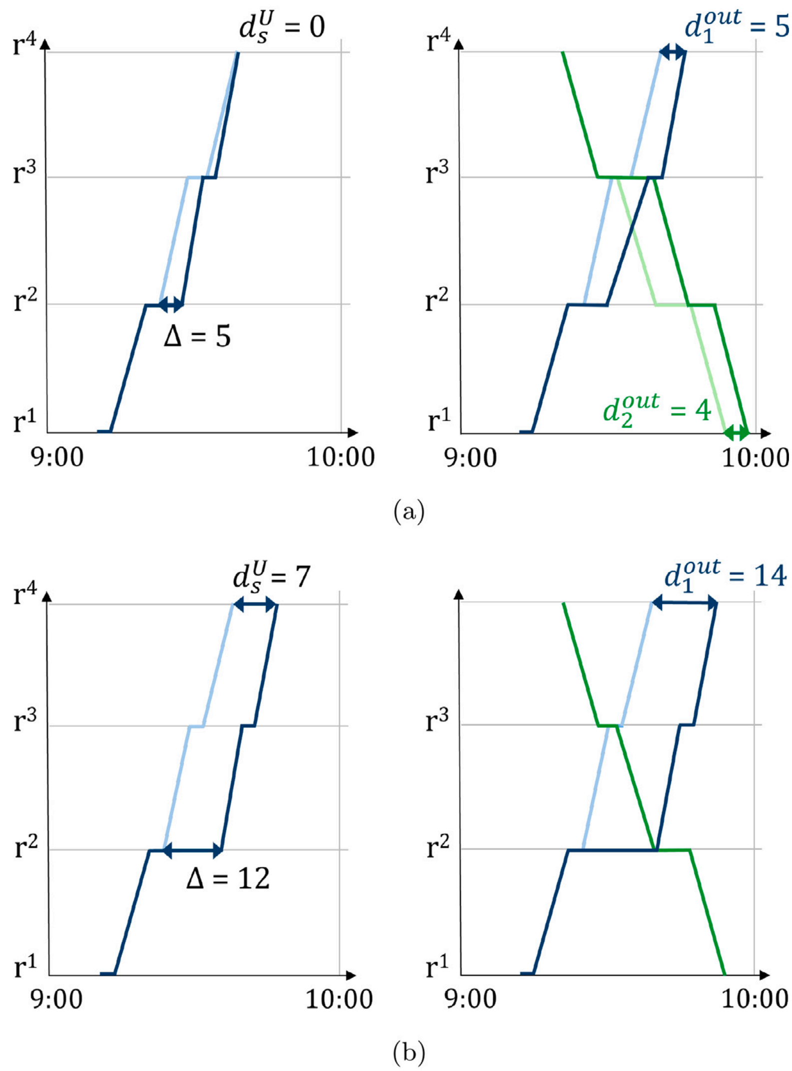



**Fig. 4.** Optimal dispatching when considering two different scenarios $(\mathbf{1}, \boldsymbol{r}^{\mathbf{2}}, \boldsymbol{\Delta})$, with (a) $\boldsymbol{\Delta} = \mathbf{5}$ and (b) $\boldsymbol{\Delta} = \mathbf{12}$ min.

$\mathcal{P}^0_{(a,r,\Delta_1)}, \mathcal{P}^1_{(a,r,\Delta_1)}, \dots, \mathcal{P}^i_{(a,r,\Delta_1)}$, until we either prove that $\mathcal{P}^i_{(a,r,\Delta_1)}$ cannot be feasibly solved, which also means that no feasible solution exists for $\mathcal{P}_{(a,r,\Delta_1)}$, or the optimal solution $\pi^i$ is deemed feasible, and thus also optimal, for $\mathcal{P}_{(a,r,\Delta_1)}$.

Since the majority of conflicts found while solving problem $\mathcal{P}_{(a,r,\Delta_{j-1})}$ are also likely to be found in $\mathcal{P}_{(a,r,\Delta_j)}$, we decide to extend the delayed row generation scheme and improve its performance, leveraging the inherent similarity between successive problem instances. Specifically, we build problem $\mathcal{P}^0_{(a,r,\Delta_j)}$ starting from the last problem solved for scenario $s = (a, r, \Delta_{j-1})$, rather than from $\mathcal{P}_{(a,r)}$ (see line 7). This ensures that all constraints in $\mathcal{P}_{(a,r,\Delta_{j-1})}$ are automatically included in the new model. We then refine and update constraints (11)–(12) to address the specific requirements of the new problem. In other words, we use $\mathcal{P}_{(a,r,\Delta_{j-1})}$ as a starting point to solve problem $\mathcal{P}_{(a,r,\Delta_j)}$. The presented approach helps to drastically reduce the computation time, by solving in a single run all scenarios associated with the same train–resource pair $(a, r)$.

## 5. Experiments

In this section, we describe the results on experiments carried out on real data from the November 2021 timetable of the Jæren line in Norway. This line, schematically shown in Fig. 5, is a railway line running along the Southern Norwegian coast from Stavanger to Moi, via Egersund, accommodating around 160 trains a day. The line is 74 km long and includes 28 stations. The infrastructure consists of double tracks from Stavanger to Skeiane, and single tracks from Skeiane to Moi, for a total of 34 tracks sections between pairs of consecutive stations. The current train service consists of both passenger (local) trains and freight trains. Local trains' services are distributed according to a fixed periodic pattern (one train every 15 min from Stavanger to Skeiane, two trains per hour from Skeiane to Nærbø, and only one train per hour all the way to Egersund).

The model described in Section 3 has been implemented in Python and solved using GUROBI 10.0.3 with default settings on an Intel(R) Xeon(R) CPU E5-2699 @ 2.2 GHz with 1.5 TB of memory.

In the railway industry, train delays are often only considered as such when greater than a certain threshold, e.g., there is a threshold of 3 min to attribute delay causes. However, most of these delays are rather smaller, and often fall below the threshold of 3 min used by the industry [26]. It is interesting to study the fragility of a timetable when assessing even such small delays. In particular, for each train–resource pair $(a, r)$ we consider the set of delay scenarios $\mathcal{S}_{ar}$ where $\Delta$ ranges from 1 to 30 min, with a time discretization of 1 min. Since most primary delays, due to congestion, appear in stations [26], influenced by variations of alighting and boarding times depending on the amount of involved passengers, or transfers between connected trains, we restrict our analysis to the subset of scenarios $\mathcal{S}_{ar}$ where resource $r$ is a station. In particular, we assume that a track inherits


**Fig. 5.** The Jæren line.

**Algorithm 1:** Computing the fragility of train–resource pair $(a, r)$

**Input:** $a$, $r$, $\{\Delta_1, \dots, \Delta_n\}$
Create problem $\mathcal{P}_{(a,r)}$ including constraints (1)–(3), (13) and objective function (14)
**for** $\Delta_j \in \{\Delta_1, \dots, \Delta_n\}$ **do**
  **if** $\Delta_j = \Delta_1$ **then**
    Create $\mathcal{P}^0_{(a,r,\Delta_1)}$ from $\mathcal{P}_{(a,r)}$ by adding constraints (11)–(12)
  **else**
    Create $\mathcal{P}^0_{(a,r,\Delta_j)}$ from $\mathcal{P}_{(a,r,\Delta_{j-1})}$ by updating constraints (11)–(12)
  $i = 0$
  **while** *Optimal solution of* $\mathcal{P}^i_{(a,r,\Delta_j)}$ *not found* **do**
    Solve $\mathcal{P}^i_{(a,r,\Delta_j)}$ to optimality
    **if** $\mathcal{P}^i_{(a,r,\Delta_j)}$ is infeasible **then**
      $\mathcal{P}_{(a,r,\Delta_j)}$ is also infeasible
      **break**
    **else**
      Let $\pi^i = (t^i, x^i, y^i, w^i)$ be the current solution
      **if** $\pi^i$ violates some of the constraints (4) and (9) not in $\mathcal{P}^i_{(a,r,\Delta_j)}$ **then**
        $i = i + 1$
        Add violated constraints (and corresponding binary variables) to $\mathcal{P}^i_{(a,r,\Delta_j)}$
      **else**
        $\pi^i$ is feasible
        $\pi^i$ is the optimal solution for $\mathcal{P}^i_{(a,r,\Delta_j)}$
        $\mathcal{P}_{(a,r,\Delta_j)} = \mathcal{P}^i_{(a,r,\Delta_j)}$
        **break**
  **end**
**end**

the fragility value associated with the preceding station. More formally, if $r^2$ is the track following station $r^1$ on the path of train $a$, and $s' = (a, r^1, \Delta)$ and $s'' = (a, r^2, \Delta)$ are the associated scenarios, we let $f(\pi^*(s''), \pi) \stackrel{\text{def}}{=} f(\pi^*(s'), \pi)$, and only compute the latter. Finally, primary delays at trains' destination stations do not cause knock-on delays, so we exclude these from the subset of scenarios as well. We set the *big-M* constant in constraints (4), (7), (8) and (10) to 1 h. We reason that, if two trains meeting in a station end up being separated there for more than an hour, it is very likely that a major breakdown, such as a partial or full blockage of a certain track section, has occurred. In this case, the problem should no longer be considered a train dispatching problem, as disruption management techniques would be a more adequate response.

For each primary delay $\Delta \in \{1, \dots, 30\}$, we have 2444 dispatching scenarios, one for each train $a \in A$ and each non-destination station $r \in R^S$ traversed by train $a$, i.e., $r \in R^S \cap R_a \backslash \{r_a^{q_a}\}$. Given the timetable and the current position of trains, when train $a$ is delayed by $\Delta$ in station $r$, the corresponding complete MILP model would be intractable for some scenarios. For example, let us consider a scenario in the afternoon, around 16:30, and look at Stavanger station, which has capacity 5. If we had 60 trains that require the use of this station, we would have to build 50 063 860 constraints (9). When we repeat this computation for 24 stations of varying capacity, it is evident that the problem becomes too difficult, even to write, for morning and afternoon scenarios. Reducing the complexity of the problem by looking at the minimal set of variables and constraints actually needed allows us to consistently solve all 30 × 2444 dispatching models.

When we use the row generation approach of Lamorgese and Mannino [5], the corresponding MILP model is able to find an optimal schedule that minimizes the considered measure of the overall delay for all scenarios in almost 12 h of computation time. Applying Algorithm 1 we are able to reduce this computational time and compute the fragility of the considered timetable in 4 h, with an average of 0.2 seconds per scenario. With this solving approach, the MILP model of a single scenario never exceeds 10431 variables and 15712 constraints. More in details, when looking at train–resource pairs requiring longest computation times, we observe that the maximum computation time required by the proposed algorithm to successfully solve the 30 dispatching scenarios related to a single pair is 82.87 seconds. This is certainly an improvement compared to the 151.03 seconds obtained by the delayed row generation scheme of [5] for the same pair. This improvement is even more striking when considering that the latter method can take up to 179.08 s in the worst-case scenario to solve all dispatching scenarios associated with a single pair, specifically pair (17, Vatnamot), where instead Algorithm 1 requires only 47.56 s. The maximum computation time of 82.87 s is associated with pair (135, Hognestad) and involves solving 252 conflicts, where 46 are station conflicts, 165 track conflicts and 41 safety margin conflicts. For the same pair, the original delayed row generation method solves a total of 2819 conflicts. Since many of those conflicts are redundant across multiple scenarios, our algorithm removes any redundancy by addressing them only once. These results certainly validate the efficacy of the proposed algorithm and our choice to improve the performance of the algorithm of [5] for the fragility computation by taking advantage of similarities between different dispatching scenarios related to the same pair.

The line plot in Fig. 6 shows the total delay distribution associated with each primary delay $\Delta \in \{1, \dots, 30\}$, computed as the number of scenarios characterized by a certain objective function value. The $x$-axis of the plot represents the number of scenarios, ordered by decreasing total delay, while the vertical axis represents the corresponding objective value. Each primary delay $\Delta$ is associated with a different color, ranging from dark green, i.e., 1 min, to dark red, i.e., 30 min. The line plot in Fig. 6 presents a large number of scenarios with small delays and a few scenarios with very large ones. Furthermore, the maximum total delay increases with the size of the primary delay $\Delta$. As a consequence, a maximum value of 1.66 h is reached when $\Delta = 30$ min. Still, the increase is not monotonous, and few exceptions can be detected due to the fragility paradox. For example, when $\Delta = 12$ min, the total delay of some scenarios is lower than the total delay corresponding to a smaller $\Delta$, e.g., $\Delta = 8$ min (visually the lime green line goes below the green line around 300 scenarios). This becomes evident when $\Delta$ is bigger than 20 min, and the orange and red lines partially remain below the yellow ones, e.g., close to 500 scenarios, which, again, is probably due to the periodicity of the considered timetable.

Fig. 7 shows the nominal timetable with the corresponding fragility values, computed as the expected value of the total delay, i.e., the sum of train delays at destination stations, obtained when $\Delta$ varies between 1 and 30 min, with a uniform probability distribution. We call this diagram *fragility map*. Note that different fragility maps might be obtained by using other probability distributions, based on the primary delays observed in practice for the specific line. Clearly, this fact does not affect the computational complexity and the significance of this study.

In Fig. 7, the $x$-axis represents the time in hours, spanning a whole day, while the $y$-axis depicts the topology of the Jæren railway line.

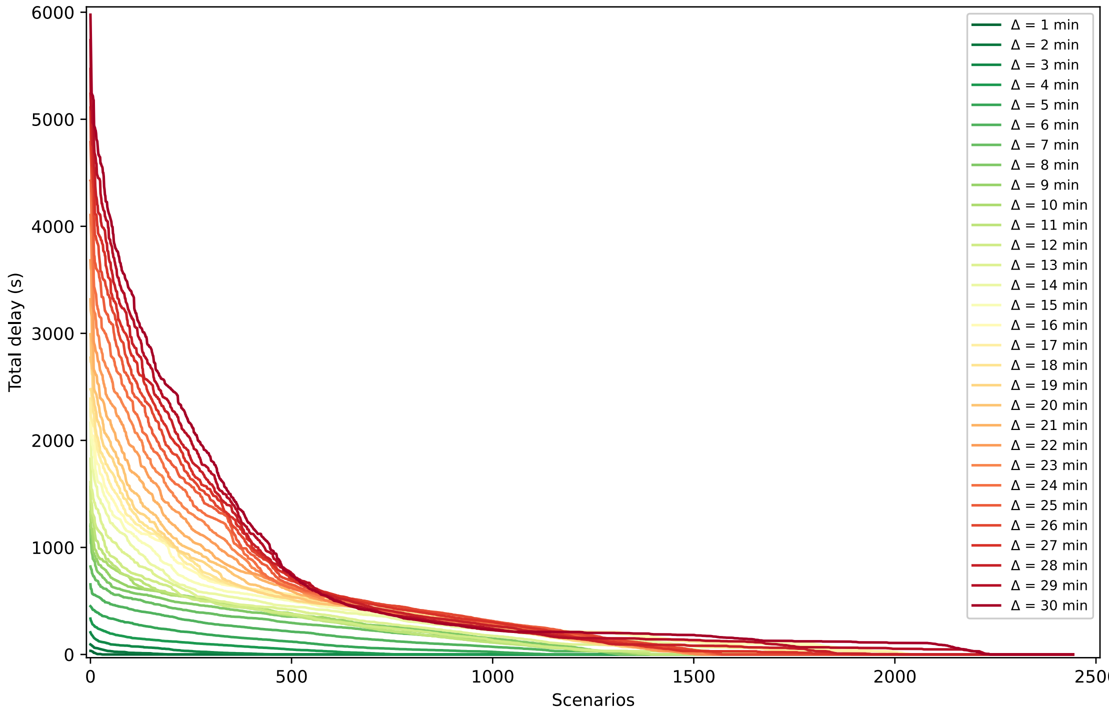



**Fig. 6.** Total delay distribution associated with primary delay $\Delta \in \{1, \ldots, 30\}$ min.


**Fig. 7.** Fragility map for the November 2021 timetable of the Jæren railway line.

Each train is shown as a single line in the graph. A color ranging from yellow to black is associated with each train–resource pair $(a, r)$, representing the corresponding expected value of the fragility, i.e., the darker the line, the more fragile that part of the timetable. As mentioned before, we associate with each track following a station on the route of a train the same fragility value of the preceding station, hence, the same color. We can see how the timetable tends to be more fragile between Ganddal and Klepp stations, i.e., at the end of the double-track sections, but also between stations Bryne and Nærbo, and Ogna and Egersund, especially when they are reached by trains going to or coming from Moi. There are some critical situations, e.g., trains departing from Stavanger at 15:24 or 18:39, but also trains departing from the same station a few minutes before every hour (at 07:54, 08:54, etc.). Since these trains travel from Stavanger up to Moi, or stop in Egersund but precede one of the trains to Moi, these particular situations are due to the scheduled pattern of these trains in the timetable.

It is interesting to compare the optimal disposition timetable of Fig. 1(b) with the fragility map in Fig. 7. The fragility of all trains going through Gausel station is small in that section of their schedule, since


**Fig. 8.** Fragility map for the November 2021 timetable of the Jæren railway line when only retiming decisions are applied.

it is possible to limit the delay propagation by optimal dispatching, as in the specific example of Fig. 1(b). However, the timetable is quite dense in Gausel and there are small margins for recovering from delays without dispatching, as shown in Fig. 1(a).

We extend the initial analysis conducted in Fig. 1 for a single scenario to the entire timetable and obtain the fragility map in Fig. 8. This considers only retiming decisions and excludes any rescheduling, thus not linking the concept of fragility to the one of recoverable robustness. In this figure, we extend the color scale in Fig. 7 to include delay values up to 11096 s. However, the common portion of the color scale is identical in both figures, enabling an easy comparison of the corresponding fragility values. As in Fig. 1, it is evident that the optimal dispatching significantly reduces the delay propagation. In this case, the expected maximum total delay, i.e., the maximum fragility in the timetable, decreases from 11096 s in Fig. 8 to 2301 in Fig. 7. The parts of the timetable with the highest recovery costs in Fig. 7, e.g., those with total delays bigger than 1500 s, do not always correspond to those in Fig. 8. Indeed, the concept of fragility has to be linked with the one of recoverable robustness, otherwise the benefits of such analysis are less evident, as more train–resource pairs appear to be critically affected by delays.

These observations confirm the importance of assessing the impact of primary delays by taking into account optimal dispatching actions, with respect to other common measures such as the amount of time margins available in the timetable between consecutive trains on a resource.

## 6. Practical applications of fragility

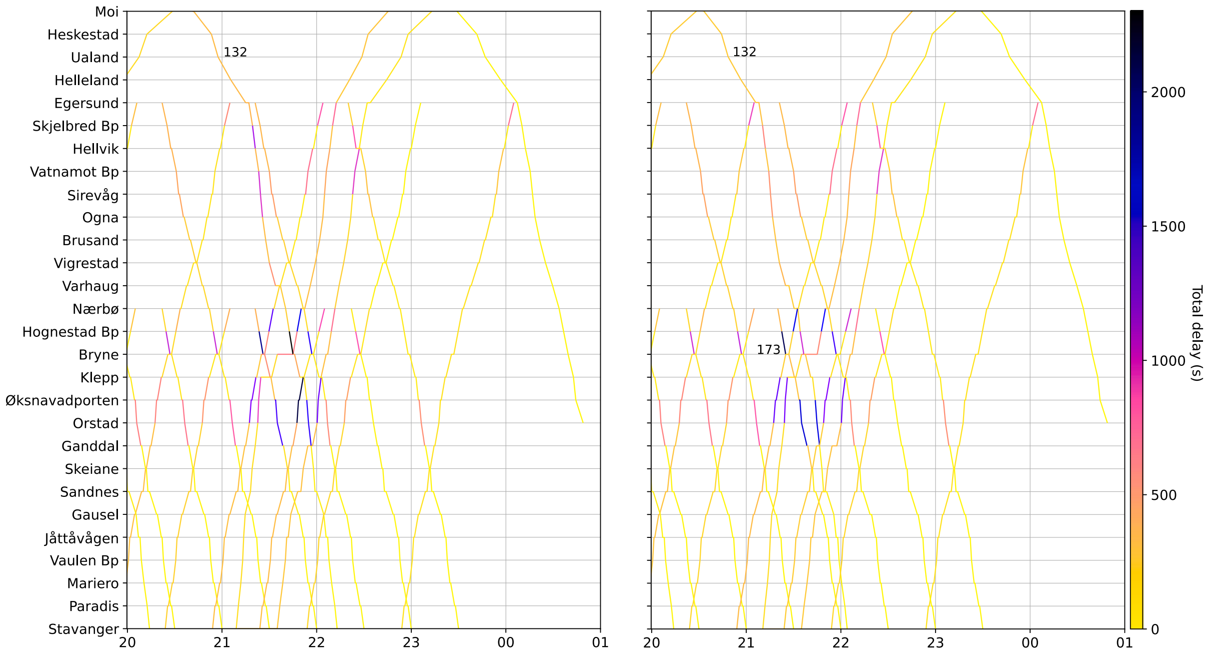
When looking at preventive approaches for disturbance management, the focus is on designing robust timetables that are able to absorb secondary delays. As discussed in the Introduction, robustness is a global measure [15,17], and therefore it is unable to highlight which regions of the railway network are less robust than others. In many cases, robustness measures are even computed without factoring in possible dispatching decisions, greatly reducing their validity in real-life settings, where such dispatching decisions are cleverly executed by expert train dispatchers.

Indeed, the use of global robustness measures may not be an issue when a timetable is perfectly periodic. However, trains often have different frequency at different times, and timetables present additional aperiodic trains or slight fluctuations (pseudo-periodicity) in their timing with no perceived impact on passengers that allow dispatchers to improve the use of infrastructure capacity. An extreme example is represented by the "slots" for trains (usually freight, but also passenger trains for special events) that seldom run. While it is necessary to include such slots in the timetable, the robustness of these trains is substantially irrelevant when compared to the ordinary trains running every day. In such cases, particularly pertinent in the European setting where the limited infrastructure capacity has to be used as best as possible, a local measure may help better understand the performances of a given timetable, thus pointing out where improvements are specifically needed. In other words, there are many practical settings where a timetable with non-uniform distribution of the fragility can be preferred to a uniform one. Clearly, it is possible to make use of a global measure of robustness in which different trains are weighted differently, thus partially taking into account this issue. However, the local measure of fragility allows managers to have a complete and better explicable picture of the timetable.

In the rest of this section, we show how the fragility can be exploited at different stages of the railway management process.

*Tactical planning.* In a typical tactical timetabling process, route planners are tasked to produce a new timetable for the next months or the next year. They usually start from an existing timetable and follow a time-consuming trial-and-error process which requires the creation of several tentative timetables to obtain a feasible one. However, given the constant increase in traffic demand, public service companies require the implementation of new effective decision support systems that are easily accessible and endorsed by practitioners at any stage of their career. Several approaches to compute robust timetables exist in the literature [27]. However, as often happens for automatic optimization tools, their tendency to provide solutions that are not easily explainable or correlated with comprehensive analysis leads to resistance in their practical application. As a result, route planners often still rely solely on their experience.

The concept of fragility tries to bridge this gap, as it can be easily incorporated as it is into robust timetabling optimization methods to




(a) (b)

**Fig. 9.** Fragility map (a) before and (b) after timetabling redesign.

address the current challenges. The intuitive nature of the fragility metric may facilitate seamless integration into existing decision-making processes, empowering practitioners to make informed adjustments with greater confidence and efficiency. Indeed, the concept of fragility represents a more practical and easy-to-understand tool that can be used by route planners within an iterative and incremental procedure (which is similar to how they ordinarily work [28]), but with a measure that can help make each iteration as effective as possible. At each iteration, the timetable may contain new, different "fragile" train–resource pairs, and the process can be iterated until a satisfactory fragility map is obtained.

To illustrate the potential of this approach and provide an initial idea of one of its possible implementation, we will now propose an example of how an iteration of this procedure would work. Fig. 9(a) shows the fragility map corresponding to the last part of the timetable in Fig. 7, i.e., from 20:00 to the end of the service. As mentioned before, the darker the color of a train–resource pair in the fragility map, the more fragile that pair is in the timetable. According to Fig. 9(a), the schedule of train 132 between stations Hognestad Bp and Bryne, i.e., pair (132, Hognestad), is clearly one of the most fragile, with a value of $F(\pi, S_{132,Hognestad}) = 2301$. To reduce the fragility of pair (132, Hognestad), for example, we may consider to let train 132 depart from station Moi 8 min earlier, and redistribute the time supplements available in the timetable to avoid conflicts, thus obtaining the timetable in Fig. 9(b). Specifically, this figure shows the fragility map corresponding to the new timetable. As in the previous section, this is computed as the expected value of the total delay obtained when $\Delta$ varies between 1 and 30 min, with a uniform probability distribution. Changing one of the most fragile parts of the timetable reduces not only the fragility of pair (132, Hognestad), from 2301 to 999 s, but also the fragility of most train–resource pairs with high recovery costs in Fig. 9(a), as evident from the change in color between the two figures, thereby improving the overall robustness of the timetable. Indeed, now the maximum fragility in the new timetable has value 2085 and it is associated to pair (173, Hognestad). A new iteration of the decision process would focus on this new fragile pair and on those that could possibly be linked to it (see Fig. 9).

It is clear that this is just a very simple example on how the fragility measure can be directly applied by practitioners as it is, without further development. Nevertheless, the practical implementation of a fragility-based timetabling approach may prove to be particularly complex in some contexts and will certainly require further in-depth study in the future.

*Strategical planning.* At this decision level, train planners make decisions on how to improve the perceived quality of train services in the long term, for example, by building new infrastructure in order to increase the capacity of the railway network, thus enabling to add more trains in the timetable and to better satisfy present and future demand. This is a complex decision process where it is particularly important to compare the effects of alternative investments on the perceived quality of service, which is significantly affected by train delays. The fragility map can play a significant role in this process, e.g., following the approach to infrastructure development driven by a long-term timetable, as in Sander and Nachtigall [29]. In fact, the most fragile sections of the long-term timetable are the prime candidates for infrastructure enhancements. For instance, the fragility map in Fig. 7 suggests to extend the double-track sections from Skeiane up to Ganddal (or even up to Nærbo), which are indeed part of the infrastructure upgrades decided by the Jernbanedirektorat of Norway for that line in future years (see Sartor et al. [19]). Moreover, Fig. 7 immediately caught the attention of the Norwegian route planners to which we presented this concept, and attributed the cause of some of the darker colors to the stations with a single capacity, e.g., Hognestad Bp and Øksnavadporten. While they always had a hunch that this was the case, they were quite satisfied to see their hypothesis verified and to be able to evaluate its impact compared to other bottlenecks.

*Operational planning (dispatching).* When a primary delay occurs during operations, dispatchers can take corrective actions to modify the nominal timetable attenuating knock-on delays. For a number of ill-case scenarios, typically involving some perturbations, dispatchers already have at hand a predefined list of actions (at Bane NOR these are called *action cards* [30]). To compute the fragility at a tactical level, we generate an optimal disposition timetable for each considered delay scenario. Since rescheduling algorithms are far from being adopted by railway companies and implemented in practice, we could exploit these disposition timetables at the operational level, and use them as action cards in case a delay scenario occurs in practice (without the need of applying any rescheduling algorithm). Indeed, a more accurate study of the fragility of a timetable allows us to shift the burden of complex timetable adjustments from real-time operations to a well-prepared tactical phase and helps in building dispatcher's trust in automated decision support systems. For instance, if scenario $(27, \text{Gausel}, 10)$ presented as an example in Fig. 1 is realized during operations, train dispatchers can decide to follow the suggested optimal dispatching decisions depicted in Fig. 1(b). As further examples, an interactive page (available at https://optimization.pages.sintef.no/train-timetable-fragility/) shows the optimal disposition timetable for each scenario $(a, r, \Delta)$ used to compute the fragility map in Fig. 7.

Fragility maps can also be exploited to increase dispatchers awareness on which sections of the railway network it is more important to drive "on-time", for instance, by alerting train drivers to carefully follow the recommended speed-profiles.

## 7. Conclusions and future research

In this paper, we introduced the concept of fragility as a new practical tool to analyze where primary delays are more likely to generate knock-on delays in a given timetable. The resulting fragility map highlights the sections of the timetable that are most prone to propagate primary delays, thus allowing a detailed analysis of the timetable.

The feedback from railway managers pointed out, on the one hand, the lack of tools to highlight critical sections of a timetable and, on the other hand, the potential of the fragility map in helping route planners to build a more robust timetable [31]. Moreover, since existing railway traffic simulators require time-consuming analysis to determine sources of delays, building a fragility map has the potential to quickly identify bottlenecks and weaknesses in a railway timetable, run what-if analyses on critical sections, and provide a valuable new perspective in addition to the current set of tools available to capacity planners [32].

We showed how the fragility concept can be exploited at different stages of the railway management process, i.e., it can be used in dispatching, to build more robust timetables, or to organize effective maintenance operations planning. This opens up a number of new interesting research directions to be explored.

Clearly, the first one involves further investigation at the operational level. For example, it would be interesting to extend the concept of fragility to disruption management, for example, by looking at a single-track failure. This would help the dispatcher evaluating more impactful decisions, such as train cancellation, train rerouting, and so on. For example, if one train in a set has to be canceled, choosing the "most fragile" could minimize the delay propagation.

The second one is linked to the automatic generation of robust timetables at a tactical level. As we are now able to explicitly compute the recovery cost of a timetable and to highlight its most critical parts, we can exploit this measure to compare alternative timetables and thus to look for the most robust timetable. For example, it is possible to look for a timetable which minimizes the recovery cost, subject to fragility upper bounds, or which minimizes the maximum fragility value, and so on. Having local measures of fragility, rather than a single global measure, gives more freedom to customize the final result.

The third direction concerns a better and optimized used of fragility maps at a strategic decision level. For example, fragility can be part of managerial decision support tools for the identification of sections of a line with over/underused capacity, or for line planning, highlighting where additional service intentions could have a higher/lower impact, or for network design, indicating where to direct additional infrastructure investments.

A fourth research direction concerns extending the concept of fragility to other contexts. In this paper the fragility has been defined with specific reference to train timetabling, however this concept can be straightforwardly applied to any job-shop scheduling problem. In fact, it could also be extended to other classes of optimization problems. Furthermore, even in the train timetabling problem there is a lot of space for further investigation. For example, the dependence of the fragility from a specific probability distribution of the disturbances, with possibly multiple delayed trains, is worth investigation. Also, the choice of different objective functions should be analyzed. In fact, there is an interesting stream of research, named delay management [14], focusing on the delay of the passengers rather than on the delay of the trains, and it would be interesting to study the fragility of a timetable from the viewpoint of the passengers.

Further research is also needed to improve the computation of the fragility map. For example, the dispatching problem solved in this paper uses deterministic data. A stochastic optimization approach would allow to include in the model the uncertainty still affecting the data, similar to what was introduced in Meloni et al. [33].

## CRediT authorship contribution statement

**M.L. Tessitore:** Writing – original draft, Visualization, Software, Methodology, Conceptualization. **G. Sartor:** Writing – original draft, Supervision, Software, Methodology, Conceptualization. **M. Samà:** Writing – original draft, Supervision, Software, Methodology. **C. Mannino:** Writing – original draft, Methodology, Conceptualization. **D. Pacciarelli:** Writing – original draft, Methodology, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This research was partially supported by SINTEF (Norway) through a Strategical Self-funded Project called RobOT (Robust Optimization for train Timetabling), the Europe's Rail Joint Undertaking (ERJU) project MOTIONAL Grant Agreement 101101973, and by the research project P20224YN23 - Co-Opetitive Sustainable Mobility Optimization (COSMO), funded by the European Union - Next Generation EU. All authors approved the version of the manuscript to be published.

## Data availability

The authors do not have permission to share data.

## References

[1] Marra AD, Corman F. From Delay to Disruption: Impact of Service Degradation on Public Transport Networks. Transp Res Rec 2020;2674(10):886–97.
[2] Bach L, Mannino C, Sartor G. MILP approaches to practical real-time train scheduling: the Iron Ore Line case.. In: INOC. 2019, p. 78–82.
[3] Corman F, D'Ariano A, Marra A, Pacciarelli D, Samà M. Integrating train scheduling and delay management in real-time railway traffic control. Transp Res Part E: Logist Transp Rev 2017;105(1):213–39.
[4] D'Ariano A, Pacciarelli D, Pranzo M. A branch and bound algorithm for scheduling trains in a railway network. European J Oper Res 2007;183(2):643–57.
[5] Lamorgese L, Mannino C. An Exact Decomposition Approach for the Real-Time Train Dispatching Problem. Oper Res 2015;63(1):48–64.
[6] Pascariu B, Samà M, Pellegrini P, D'Ariano A, Rodriguez J, Pacciarelli D. Effective train routing selection for real-time traffic management: Improved model and ACO parallel computing. Comput Oper Res 2022;145:105859.
[7] Tessitore ML, Samà M, D'Ariano A, Hélouët L, Pacciarelli D. A simulation-optimization framework for traffic disturbance recovery in metro systems. Transp Res Part C: Emerg Technol 2022;136:103525.

[8] Liebchen C, Lübbecke M, Möhring R, Stiller S. The concept of Recoverable Robustness, Linear Programming Recovery, and Railway Applications. In: Robust and online large-scale optimization. Springer; 2009, p. 1–27.
[9] Fischetti M, Salvagnin D, Zanette A. Fast approaches to improve the robustness of a railway timetable. Transp Sci 2009;43(3):321–35.
[10] Goverde RM. Railway Timetable Stability Analysis Using Max-plus System Theory. Transp Res Part B: Methodol 2007;41(2):179–201.
[11] Kroon LG, Dekker R, Vromans MJ. Cyclic Railway Timetabling: a Stochastic Optimization Approach. In: Algorithmic methods for railway optimization. Springer; 2007, p. 41–66.
[12] Salido MA, Barber F, Ingolotti L. Robustness for a single railway line: Analytical and simulation methods. Expert Syst Appl 2012;39(18):13305–27.
[13] Yin J, Yang L, D'Ariano A, Tang T, Gao Z. Integrated backup rolling stock allocation and timetable rescheduling with uncertain time-variant passenger demand under disruptive events. INFORMS J Comput 2022;34(6):3234–58.
[14] Cicerone S, Di Stefano G, Schachtebeck M, Schöbel A. Multi-stage recovery robustness for optimization problems: A new concept for planning under disturbances. Inform Sci 2012;190:107–26.
[15] Goerigk M, Schöbel A. Recovery-to-optimality: A new two-stage approach to robustness with an application to aperiodic timetabling. Comput Oper Res 2014;52:1–15.
[16] Grafe V, Schöbel A. Recoverable robust periodic timetabling. In: 23rd symposium on algorithmic approaches for transportation modelling, optimization, and systems (ATMOS 2023). Schloss Dagstuhl-Leibniz-Zentrum für Informatik; 2023.
[17] Cacchiani V, Caprara A, Galli L, Kroon L, Maróti G, Toth P. Railway rolling stock planning: Robustness against large disruptions. Transp Sci 2012;46(2):217–32.
[18] Andersson EV, Peterson A, Törnquist Krasemann J. Quantifying railway timetable robustness in critical points. J Rail Transp Plan Manag 2013;3(3):95–110.
[19] Sartor G, Mannino C, Nygreen T, Bach L. A MILP model for quasi-periodic strategic train timetabling. Omega 2023;116:102798.
[20] Pachl J, White T. Efficiency through integrated planning and operation. In: Implementation of heavy haul technology for network efficiency, international heavy haul association, proceedings, virginia beach. 2003, p. 6–69.
[21] Leutwiler F, Corman F. A logic-based Benders decomposition for microscopic railway timetable planning. European J Oper Res 2022.
[22] Fischetti M, Monaci M. Using a general-purpose mixed-integer linear programming solver for the practical solution of real-time train rescheduling. European J Oper Res 2017;263(1):258–64.
[23] Meng L, Zhou X. Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. Transp Res Part B: Methodol 2011;45(7):1080–102.
[24] Pellegrini P, Pesenti R, Rodriguez J. Efficient train re-routing and rescheduling: Valid inequalities and reformulation of RECIFE-MILP. Transp Res Part B: Methodol 2019;120:33–48.
[25] Zhang H, Li S, Wang Y, Wang Y, Yang L. Real-time optimization strategy for single-track high-speed train rescheduling with disturbance uncertainties: A scenario-based chance-constrained model predictive control approach. Comput Oper Res 2021;127:105135.
[26] Palmqvist C-W. Delays and timetabling for passenger trains (Ph.D. thesis), Lund University; 2019.
[27] Caimi G, Kroon L, Liebchen C. Models for railway timetable optimization: Applicability and applications in practice. J Rail Transp Plan Manag 2017;6(4):285–312.
[28] Kloster O, Luteberget B, Mannino C, Sartor G. An optimization-based decision support tool for incremental train timetabling. Oper Res Forum 2023;4.
[29] Sander T, Nachtigall K. Introducing an Optimization Model for Timetable Based Railway Network Design. In: 11th triennial symposium on transportation analysis conference (TRISTAN XI), June 19-25, 2022, mauritius island. 2022.
[30] Bane NOR. Bane NOR action cards. 2021, Last Visited: 10 May, 2023.
[31] Pettersen K. Personal communication. 2024.
[32] Nygreen T. Personal communication. 2024.
[33] Meloni C, Pranzo M, Samà M. Risk of delay evaluation in real-time train scheduling with uncertain dwell times. Transp Res Part E: Logist Transp Rev 2021;152:102366.