Applied Mathematical Modelling 52 (2017) 255–273

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

A decision making procedure for robust train rescheduling based on mixed integer linear programming and Data Envelopment Analysis

Graziana Cavone [a], Mariagrazia Dotoli [b], Nicola Epicoco [b], Carla Seatzu [a,*]

[a] Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy
[b] Department of Electrical and Information Engineering, Polytechnic of Bari, 70125 Bari, Italy

ARTICLE INFO

Article history:
Received 26 January 2017
Revised 26 May 2017
Accepted 17 July 2017
Available online 26 July 2017

Keywords:
Decision making
Railways
Real-time
Rescheduling
Robustness
Data Envelopment Analysis

ABSTRACT

This paper presents a self-learning decision making procedure for robust real-time train rescheduling in case of disturbances. The procedure is applicable to aperiodic timetables of mixed-tracked networks and it consists of three steps. The first two are executed in real-time and provide the rescheduled timetable, while the third one is executed offline and guarantees the self-learning part of the method. In particular, in the first step, a robust timetable is determined, which is valid for a finite time horizon. This robust timetable is obtained solving a mixed integer linear programming problem aimed at finding the optimal compromise between two objectives: the minimization of the delays of the trains and the maximization of the robustness of the timetable. In the second step, a merging procedure is first used to join the obtained timetable with the nominal one. Then, a heuristics is applied to identify and solve all conflicts eventually arising after the merging procedure. Finally, in the third step an offline cross-efficiency fuzzy Data Envelopment Analysis technique is applied to evaluate the efficiency of the rescheduled timetable in terms of delays minimization and robustness maximization when different relevance weights (defining the compromise between the two optimization objectives) are used in the first step. The procedure is thus able to determine appropriate relevance weights to employ when disturbances of the same type affect again the network. The railway service provider can take advantage of this procedure to automate, optimize, and expedite the rescheduling process. Moreover, thanks to the self-learning capability of the procedure, the quality of the rescheduling is improved at each reapplication of the method. The technique is applied to a real data set related to a regional railway network in Southern Italy to test its effectiveness.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

The efficient management of railway traffic is crucial for rail companies to provide their customers with a quality service [1]. In particular, both companies and customers are interested in an on-time service. The first can avoid sanctions applied if the accumulated delay overcomes the maximum imposed by contract, while the latter can benefit from a reliable service

* Corresponding author.
E-mail addresses: graziana.cavone@diee.unica.it (G. Cavone), mariagrazia.dotoli@poliba.it (M. Dotoli), nicola.epicoco@poliba.it (N. Epicoco), seatzu@diee.unica.it (C. Seatzu).

http://dx.doi.org/10.1016/j.apm.2017.07.030
0307-904X/© 2017 Elsevier Inc. All rights reserved.

without loss of money and time. Moreover, an efficient service can improve customers' loyalty to the company and this produces, as a secondary effect, a reduction of road traffic in favour of a more sustainable railway transport.

Automated real-time rescheduling emerges as a promising technique to manage railway traffic when unexpected events affect the normal behaviour of the network. Basically, train rescheduling consists in retiming the offline scheduled traffic (i.e., the nominal timetable) to minimize undesired effects on the railway service (e.g., train delays, customer discomfort, energy consumption) [2–6]. Typically, unpredictable events are distinguished into disturbances (i.e., relatively small perturbations such as signal malfunctions or no-show of staff) and disruptions (i.e., large and particularly damaging external accidents such as trains or infrastructure breakdowns) [7–9]. Both cause the nominal timetable to become invalid because at least one train deviates from its original schedule. Our contribution refers to the more common disturbances.

Generally, Train Dispatchers (TDs) manage disturbances mostly manually based on their experience and knowledge [2,7,10]. However, this highly time-consuming approach normally leads to suboptimal timetables. Such a manual procedure can be refined and speeded up by using automated real-time rescheduling procedures that may support TDs in determining in real-time suitable control actions and updating timetables while optimizing some traffic performance indices [2,7,10]. Furthermore, in order to improve the quality of the automated rescheduling, it is important to evaluate the corresponding outputs, predict the consequences of different control actions, and keep trace of the most favourable one. In this paper, we achieve this aim by including into a suitable decision making procedure both an automated real-time rescheduling process and an offline self-learning technique. Such a train rescheduling procedure can on the one hand reduce the workload of the TD, by automatically and autonomously performing a real-time rescheduling, and on the other hand improve the effectiveness of the rescheduling at each reapplication of the method.

More in detail, the presented decision making procedure is valid for railway systems using aperiodic timetables and presenting mixed (single- and double-) tracked networks. We do not consider any limitation on the dimension and the topology of the system. The procedure consists of three steps: in the first one, an optimal timetable over an appropriate Time Horizon (*TH*) is obtained by solving a Mixed Integer Linear Programming (MILP) problem aimed at finding the best compromise between the minimization of delays of trains and the maximization of the robustness of the timetable. In the second step, a merging procedure is first used to join the optimal timetable over the chosen *TH* with the nominal one in the remaining time window of the timetable. Then, all possible conflicts arising after *TH* are iteratively solved by means of a heuristic procedure, which calculates a near-optimal rescheduling solution. Finally, the third step consists in an offline self-learning procedure aimed at predicting the results and effectiveness of alternative control actions and updating an external database with the most appropriate solution to use in the event that a disturbance of the same type occurs in the future. To this aim, a cross-efficiency fuzzy Data Envelopment Analysis (DEA) method proposed by some of the authors for the performance evaluation of healthcare systems in [11] is applied to determine the efficiency of the rescheduled timetable (in terms of reduction of delays and maximization of robustness) and to rank alternatives according to their efficiency values.

The proposed decision making procedure is tested on a real case study that is a portion of a regional railway network where a set of trains is affected by a disturbance at a rush hour in a weekday. The railway network is located in Southern Italy and is constituted mostly by single tracks with few double track segments, and such that in some stations only a train can stop or pass through.

To confirm the effectiveness of the proposed procedure we compare the outcomes of the technique with those obtained with the traditional TD manual procedure. Moreover, we accurately validate the effects of the robustness maximization by statistically evaluating two indices: the number of conflicts of trains and the time delay caused by the occurrence of an additional disturbance both for the robust rescheduled timetable and for a rescheduled timetable obtained by only minimizing delays.

Summarizing, the main contributions of this paper consist in:

(1) Presenting an automatic decision making procedure for robust real-time train rescheduling in case of disturbances.
(2) Stating and solving a MILP problem for mixed-tracked railway systems that simultaneously addresses the minimization of the delays caused by a disturbance and the maximization of the robustness of the timetable.
(3) Integrating the resolution of the MILP problem with a heuristic procedure to speed-up the real-time rescheduling procedure.
(4) Proposing an offline DEA-based self-learning procedure, which predicts and evaluates the consequences of alternative control actions to improve the effectiveness of the method at each reapplication.

The following subsection compares the manuscript contributions with the current state of the art and discusses the main managerial implications of the approach.

1.1. Positioning of the approach with respect to the state of the art and managerial implications

In the related literature, only few studies have been developed in the context of decision making procedures for real-time train rescheduling (see for instance the discussions in [12] and [13]). These contributions mainly focus on automating the rescheduling procedure, while there is still a lack of self-learning approaches that can strongly improve the performance of the decision making and quickly predict the results of changes in control actions. As a matter of fact, nowadays railway companies are still seeking for automatic solutions to improve traffic management in order to enhance timeliness and reliability of railway services.

Exhaustive discussions on automated real-time rescheduling can be found in [6,14,15], and [16], showing that, in the relevant literature, three main classes of computer-based rescheduling approaches can be broadly identified: simulation models, heuristic procedures, and mathematical optimization models, or a combination of them. The most common framework to reschedule railway traffic is MILP [1,16,17].

Traditionally, in the related literature, rescheduling optimization models consider a single-objective function. However, the nature of the train rescheduling problem is intrinsically multi-objective due to multiple conflicting interests of the involved stakeholders and to social and environmental issues. Hence, multi-objective approaches generally produce better rescheduling alternatives. To actually provide a significant support to the TD, rescheduling approaches in railway networks should at least possess two features, namely, timeliness and robustness [15]. The former consists in the minimization of delays caused by the occurrence of a disturbance, while the latter is the ability of the timetable to absorb such disturbances, to tolerate a certain degree of uncertainty, and/or to cope with unexpected troubles without significant modifications [18–20].

When disturbances arise, they cause primary delays to the directly affected train that typically propagate to other trains as secondary delays. Consequently, one of the major requirements of an effective rescheduling is to limit the spread over the network of secondary delays, which in turn requires that: (1) primary delays can be absorbed, (2) few primary delays result in small secondary delays, and (3) secondary delays can quickly disappear thanks to simple dispatching operations, without spreading over the network [21,22].

Usually, robustness is achieved by adding some time reserves in the timetable to allow for flexibility when rescheduling traffic to prevent delays' further spreading. Such time reserves can be classified into recovery times and buffer times. The former are time reserves computed offline by increasing the travel time with respect to the minimum one (i.e., traveling at the maximum speed), while the latter are time reserves over the minimum separation time between consecutive train paths [6,23]. In other words, recovery times are introduced mainly to reduce primary delays, while buffer times are defined to limit the propagation of secondary delays. Most studies ensure the robustness of timetables to small disturbances in the offline scheduling process by perturbing the nominal timetable with observed or simulated disturbances (see for instance [10,19,24–28], and [29]). On the contrary, as reported in [19], few contributions aim at increasing the timetable robustness in the real-time rescheduling process, although it can be more effective than the offline process, which is an open-loop control process. Such a lack is mainly due to the high combinatorial complexity of multi-objective MILP problems that often make computation times too high for the real-time requirements. In order to overcome such a limitation, a good compromise can be achieved by combining the optimal rescheduling by MILP techniques with a heuristics that simplifies and speeds up the rescheduling procedure [14].

Another important aspect to take into account is that of iteratively evaluating and improving the effectiveness of the real-time rescheduling. To this aim, due to the recalled multi-objective nature of the problem and thanks to the large amount of available data, Multi-Criteria Decision Making (MCDM) techniques can represent an efficient tool. In fact, in the last decades, the use of MCDM methods has extensively increased in different application areas (see the review in [30]). Several MCDM approaches are available, each with its own advantages and disadvantages [30]. Most MCDM techniques are usually applied prior to decision making or project execution, while DEA is more often utilized for the evaluation of schemes already implemented [31]. The ease of use of the DEA and its ability to quantify results make this technique an efficient tool for simplifying data analysis. Moreover, thanks to the possibility of combining DEA with the fuzzy set theory (see [11]), the TD may take into account uncertainties affecting the rescheduling process.

In the above context, this paper presents a three-steps decision making procedure, which is characterized by three main novelties that are able to significantly alleviate the TD real-time work: (1) a rescheduling technique consisting in solving a bi-objective MILP optimization problem with the twofold aim of minimizing delays and maximizing the robustness of the timetable, (2) a heuristics to speed up the rescheduling process so as to cope with the real-time requirements, (3) an offline self-learning procedure based on the fuzzy DEA technique to evaluate and rank the effectiveness of alternative rescheduling actions.

The MILP approach that we present in the first step of the decision making procedure is formulated on the basis of the techniques in [10] and in [32]. In particular, we still consider the railway traffic as a sequence of events that can be assigned when trains, technical, and operator constraints are fulfilled. However, here we present a bi-objective formulation that, as already stated, aims at minimizing delays at each station and maximizing the overall robustness of the timetable. With respect to other existing contributions (see also [2,7,14,15,33]), here the focus is on preventively reducing the impact of additional disturbances by incorporating the robustness maximization directly in the rescheduling problem statement, rather than providing a rescheduling procedure coping with disturbances after their occurrence.

Furthermore, with respect to the train rescheduling approaches presented in other works (e.g., in [10,12], and [13]), we combine the simplicity of heuristics -to reduce the computational complexity- with the precision of MILP.

In addition, our decision making procedure provides the offline self-learning procedure to allow a performance evaluation of the obtained solution. To the best of our knowledge, the DEA technique has been used in the railway scheduling context only in [34], and in a setting very different from the present one. In fact, contrarily to [34], we consider the DEA technique in a fuzzy setting to cope with data imprecision and uncertainty and we focus on the robustness concept, which is disregarded in [34].

Summing up, the procedure is useful for railway companies (to provide their customers with on-time services, reduce sanctions or penalties, and avoid possible errors caused by a manual rescheduling) and for passengers (to reduce waiting times and delays or to limit travel discomforts).

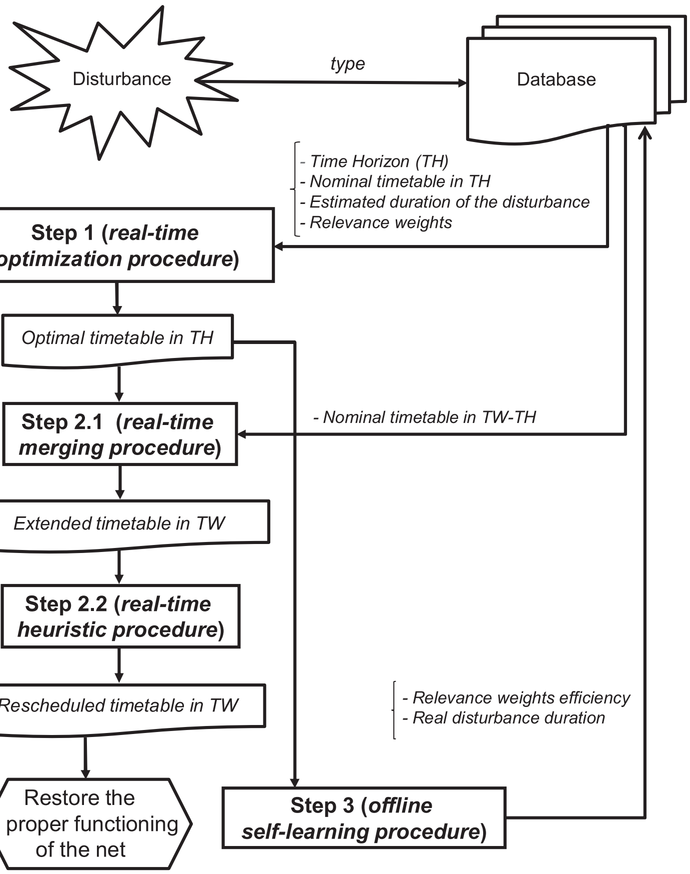

Fig. 1. A flow-chart representation of the decision making procedure.

1.2. Structure of the paper

The paper remainder is organized as follows. Section 2 presents the architecture of the proposed decision making procedure and details its steps. Section 3 presents the real case study and illustrates the experimental results. Section 4 summarizes the concluding remarks and points out the future lines of our research in this framework. Finally, Appendix A provides an explanation of the constraints used in the MILP problem formulation of Step 1, while Appendix B provides a description of the cross-efficiency fuzzy Data Envelopment Analysis method used in Step 3.

2. The proposed decision making procedure

The proposed decision making procedure is schematically illustrated in the flowchart in Fig. 1 and consists of three main steps.

Once a disturbance occurs, the TD queries the database that contains the following information:

1) The *Time Horizon* (*TH*) in which the optimization procedure (Step 1) has to be performed. This value depends on the current disturbance and on the network complexity and infrastructure.
2) The *nominal timetable in TH*, namely the timetable originally scheduled offline over the time horizon.
3) An estimate of the duration of the disturbance calculated using statistical methods, as usually happens in railway networks [33], or other more sophisticated techniques (see for instance [35]).
4) The most suitable *relevance weights* to be used in the objective function of the optimization procedure in Step 1. Again, these values are provided by the self-learning procedure in Step 3.

After the database querying, the real-time optimization procedure in Step 1 is executed (see Section 2.1 for more details), obtaining the *optimal timetable in TH* as the best compromise between two objectives (suitably weighted by the relevance weights): the minimization of delays of trains and the maximization of the robustness of the timetable. The resulting timetable is guaranteed to be adherent to the nominal timetable, conflict-free (i.e., without unfeasible train coincidences in stations or crossings at single-track lines) within *TH*, and robust with respect to additional small disturbances.

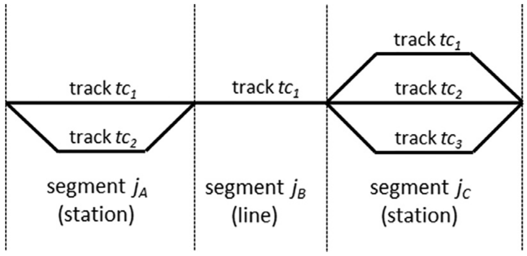

Fig. 2. An example of a generic railway line (adapted from [2]).

The real-time merging procedure in Step 2.1 (see Section 2.2) provides an *extended timetable in* the whole scheduling Time Window (*TW*). This is done combining the optimal timetable in *TH* with the nominal one in the remaining time horizon of length *TW*–*TH* (also obtained by the database queering). Note that the time interval *TW* is the overall time window for which the aperiodic rescheduling has to be performed, so it is at most one day. Obviously, the absence of conflicts is only guaranteed in *TH*, which cannot be too large in order to ensure an optimal rescheduling in real-time. Therefore, an analysis of conflicts eventually arising in the remaining interval of length *TW*–*TH* is performed. This is done at Step 2.2 via the real-time heuristic procedure detailed in Section 2.3, which allows obtaining in real-time a conflict-free *rescheduled timetable in TW*.

Finally, Step 3 is applied. It consists in an offline self-learning procedure aimed at updating the database in order to better select the relevance weights in the case of future applications of the technique for disturbances of the same type. Indeed, since the decision making procedure is based on estimates of the duration of the disturbance, it is important to assess the effectiveness of the adopted solution, so as to obtain a more accurate rescheduling in the event that a disturbance of the same type occurs again. To this aim, Step 3 of the presented technique applies a cross-efficiency fuzzy Data Envelopment Analysis [11] to evaluate the efficiency of the obtained solution (in terms of delay at each station and robustness of the timetable) compared to that of other possible solutions obtained by varying the relevance weights in the objective function of the optimization procedure (see Section 2.4 for more details on this step). As output of Step 3, the most suitable relevance weights along with the type and duration of the real disturbance are provided and stored in the database.

An important remark should be made. The real-time rescheduling is based on real-time information about the railway network state. The TD is continuously updated about the evolution of the system. Therefore, if the duration of the real disturbance is overestimated, then the rescheduled timetable is still valid. On the contrary, if the disturbance is underestimated, then, thanks to the short computation times of the whole technique, the TD can restart the procedure with updated information about the duration of the disturbance and the state of the network, thus obtaining a more appropriate rescheduling.

2.1. Step 1: the real-time optimization procedure

Step 1 aims at finding, in the given time horizon *TH*, the optimal compromise between the minimization of delays in the railway network, due to the occurrence of a generic disturbance, and the maximization of the robustness of the resulting timetable to secondary delays and to the occurrence of additional disturbances. This is achieved by stating and solving a MILP problem. Before introducing the mathematical formulation of the problem, let us briefly recall some background on railway networks and provide the basic notation used in the rest of the paper (see also Table 1).

Railway networks are usually divided into connected segments, which can be of two types (see Fig. 2): line segments and station segments [2, 16]. The former include tracks linking stations, the latter include tracks in stations. Each segment can be composed by single or parallel tracks, and each track can be occupied by at most one train at a time.

As an example, Fig. 2 provides the representation of a generic infrastructure: station A (segment j_A) comprises two tracks (tc_1 and tc_2), station C (j_C) three tracks (tc_1, tc_2, tc_3), and they are connected by a line segment (i.e., a rail connection) j_B, which is constituted by a single track.

The optimization procedure in Step 1 is based on the notion of *event*, which is a train request to occupy a track for a well-defined time interval. The occurrence of a disturbance deviates the involved train from its nominal behaviour. This implies that the end of the affected event has to be necessarily shifted according to the duration of the occurred disturbance before the application of Step 1. As already specified, the primary delay causes secondary delays on the other trains in the timetable. In this paper, we call *delay of the generic event k* (and denote it z_k) the difference between the end of the event after its rescheduling (due to the occurrence of a disturbance) and its nominal end. Furthermore, we call $Buff_k$ the *buffer time* added to event *k*, i.e., an extra time suitably added to the estimated duration of event *k* corresponding to stop in station.

By an analogy with traditional risk management procedures [36], the primary aim of introducing buffer times in the timetable is to increase the flexibility and achieve a specific goal, i.e., to let the network absorb secondary delays and possible additional disturbances. However, the introduction of buffer times can further increase the delay of events associated with a stop in station and ultimately the delay propagation in the whole network. For this reason, in order to provide a better service for passengers, the first goal of the optimization procedure is to minimize what we call *cumulative delay*,

Table 1
Summary of the notation.

Parameters	Physical meaning
T	Set of trains to be rescheduled in TH
i	Generic train in T
g_i^{train}	Length of train $i \in T$
B	Set of segments of the network
j	Generic segment in B
Pj	Set of tracks composing segment j
tc	Generic track in P_j
$g_{j,tc}^{track}$	Length of generic track tc of segment j
E	Set of events related to all trains in TH
k	Generic event in E
$Ki \subseteq E$	Ordered set of events of the generic train $i \in T$ (according to the nominal timetable)
n_i	Last event in K_i
$L_j \subseteq E$	Ordered set of events of the generic segment $j \in B$ (according to the nominal timetable)
$E^{connection} \subseteq E \times E$	Set of couples $(k, \hat{k})$formed by connected events (i.e., $\hat{k}$ must not start until k has ended)
$g_{k,\hat{k}}^{connection}$	Minimum exchange time between connected events
h_k	Binary variable indicating whether event k corresponds to a 'stop in station' or not
$R_{i,k}$	Robustness index of train i within TH with respect to event k
(α,β)	Relevance weights
$Flow_k$	Average percentage of passengers getting on train involved at a 'stop in station' event k
TT_k	Percentage of tightness of track at event k
$NSucT_k$	Number of trains that can be perturbed by the delayed train after the event k in TH
d_k	Nominal duration of event k
d_k^{trip}, d_k^{stop}	Minimum nominal duration for trip and stop times of event k
δ_k	Recovery time for event k in the nominal timetable
$b_k^{nominal}, e_k^{nominal}$	Start and end times of event k in the nominal timetable
$b_k^{static}, e_k^{static}$	Start and end times of event k that has already begun when the disturbance occurs
o_k	Point of origin of event k, indicating whether trains associated with events k and k+1 are traveling or not in the same direction
Δ_j^M	Safety time between two trains travelling in opposite directions at segment j
Δ_j^F	Safety time between two trains travelling in the same direction at segment j
M	Large positive constant (i.e., the length of the largest possible TH)
w_k	Fixed threshold to activate the binary variable ε_k
$Buff_{\max}$	Upper bound of variable $Buff_k$, $k \in E$
Decision variab.	Physical meaning
z_k	Real variable indicating the delay of the generic event k
$Buff_k$	Real variable indicating the buffer time associated with the generic 'stop in station' event k
x_k^{begin}	Real variable indicating the start time of event k
x_k^{end}	Real variable indicating the end time of event k
$q_{k,j,tc}$	Binary variable indicating whether event k uses track tc of segment j or not
$\lambda_{k,\hat{k}}$	Binary variable indicating whether event k is rescheduled to occur after $\hat{k}$
$\gamma_{k,\hat{k}}$	Binary variable indicating whether event k occurs before $\hat{k}$ as in the nominal timetable
ε_k	Binary variable indicating if train i reaches (or not) the stop in station event k with a delay larger than w_k time units

which is defined as the sum on the whole network of the extended delays. The extended delay of an event is calculated as the sum of the delay associated with the event and the corresponding buffer time, when the event is a stop in station, otherwise it corresponds to the delay of the event. The second goal is to minimize the effect of possible additional disturbances, which is realized as explained before, by adding proper buffer times to the events corresponding to stop in station. Their effect on the robustness of the obtained timetable can be quantified by a suitable index R, which has to be maximized [19,25].

Summarizing, given the set of trains T (whose generic element is indexed by i), the set of events E (whose generic element is indexed by k), and being $K_i \subseteq E$ the ordered set of events associated with train i, the objective function aims at minimizing the cumulative delay of the network while maximizing its robustness:

$$Min\left[\alpha \cdot \sum_{k \in E} (z_k + Buff_k \cdot h_k) - \beta \cdot R\right]. \quad (1)$$

In Eq. (1) α and β represent the relevance weights assigned to normalize and balance the two terms of the objective function, and these two parameters are chosen based on the TD preferences stored in the database or according to the suggestions resulting from Step 3. Furthermore, the binary parameter h_k specifies whether the event k is associated with a station segment ($h_k = 1$) or with a line segment ($h_k = 0$). In fact, the buffer time $Buff_k$ can be added to the delay z_k only in case of a stop in station event. Therefore, $(z_k + Buff_k \cdot h_k)$ represents the extended delay. The robustness of the network is quantified in (1) by index R defined as follows [25]:

$$R = \sum_{i=1}^{|T|} \sum_{k=1}^{|K_i|} R_{i,k}, \quad \text{where } R_{i,k} = Buff_k \cdot Flow_k \cdot TT_k \cdot NSucT_k \cdot \frac{(|K_i| - k)}{|K_i|} \cdot h_k, \text{ for } k = 1, \dots, |K_i|. \tag{2}$$

In Eq. (2) $Flow_k$ is the average percentage of travelers getting on the ith train at event k. Note that, given an event k according to the considered notation, it is univocally determined which is the train involved in it. According to [25], parameter $TT_k = (|K_i| - (k-1))/|K_i|$ is the percentage of tightness of tracks between stations, where the tightest track is defined as the longest distance track, i.e., given $|K_i|$ events for the ith train, the tightest track has $TT_1 = 1$, the second tightest track has $TT_2 = (|K_i| - 1)/|K_i|$, the third one has $TT_3 = (|K_i| - 2)/|K_i|$, and so on. The parameter $NSucT_k$ is the number of trains in the time horizon that could be perturbed by the delayed train i after the event k.

The minimization of (1) is performed under the following constraints:

Train constraints:

$$x_k^{end} = x_{k+1}^{begin}, \quad \forall k \in K_i - \{n_i\}, i \in T \tag{3}$$

$$x_k^{end} - x_k^{begin} - Buff_k \cdot h_k \geq d_k - \varepsilon_k \delta_k, \text{ if } d_k > \varepsilon_k \delta_k, \quad \forall k \in E$$
$$\text{where} \begin{cases} \delta_k = 0 \text{ if}(d_k < d_{k_{trip}}, h_k = 0) \vee (d_k < d_{k_{stop}}, h_k = 1) \\ \delta_k > 0 \text{ if}(d_k \geq d_{k_{trip}}, h_k = 0) \vee (d_k \geq d_{k_{stop}}, h_k = 1) \end{cases} \text{and} \begin{cases} \varepsilon_k = 1 \text{ if} z_k + Buff_k > w_k \\ \varepsilon_k = 0 \text{ otherwise} \end{cases} \tag{4}$$

$$x_k^{begin} \geq b_k^{\text{nominal}}, \quad \forall k \in E : h_k = 1 \tag{5}$$

$$x_k^{begin} = b_k^{static}, \forall k \in E : \quad b_k^{static} > 0 \tag{6}$$

$$x_k^{end} = e_k^{static}, \forall k \in E : \quad e_k^{static} > 0 \tag{7}$$

$$x_k^{end} - Buff_k \cdot h_k - e_k^{\text{nominal}} \leq z_k, \quad \forall k \in E \tag{8}$$

Technical constraints:

$$\sum_{tc \in P_j} q_{k,j,tc} = 1, \quad \forall k \in L_j, j \in B \tag{9}$$

$$q_{k,j,tc} + q_{\hat{k},j,tc} - 1 \leq \lambda_{k,\hat{k}} + \gamma_{k,\hat{k}}, \quad \forall k, \hat{k} \in L_j, tc \in P_j, j \in B; k < \hat{k} \tag{10}$$

$$x_{\hat{k}}^{begin} - x_k^{end} \geq \Delta_j^M \gamma_{k,\hat{k}} - M(1 - \gamma_{k,\hat{k}}), \quad \forall k, \hat{k} \in L_j, j \in B; k < \hat{k}, o_{\hat{k}} \neq o_k \tag{11a}$$

$$x_{\hat{k}}^{begin} - x_k^{end} \geq \Delta_j^F \gamma_{k,\hat{k}} - M(1 - \gamma_{k,\hat{k}}), \quad \forall k, \hat{k} \in L_j, j \in B; k < \hat{k}, o_{\hat{k}} = o_k \tag{11b}$$

$$x_k^{begin} - x_{\hat{k}}^{end} \geq \Delta_j^M \lambda_{k,\hat{k}} - M(1 - \lambda_{k,\hat{k}}), \quad \forall k, \hat{k} \in L_j, j \in B; k < \hat{k}, o_{\hat{k}} \neq o_k \tag{12a}$$

$$x_k^{begin} - x_{\hat{k}}^{end} \geq \Delta_j^F \lambda_{k,\hat{k}} - M(1 - \lambda_{k,\hat{k}}), \quad \forall k, \hat{k} \in L_j, j \in B; k < \hat{k}, o_{\hat{k}} = o_k \tag{12b}$$

$$\lambda_{k,\hat{k}} + \gamma_{k,\hat{k}} \leq 1 \quad \forall k, \hat{k} \in L_j, j \in B; k < \hat{k} \tag{13}$$

$$g_i^{train} q_{k,j,tc} \leq g_{j,tc}^{track} \quad \forall k \in (K_i \cap L_j), tc \in P_j, j \in B, i \in T \tag{14}$$

Operator preferences:

$$z_k + Buff_k \cdot h_k - w_k \leq M\varepsilon_k \quad \forall k \in E \tag{15}$$

$$x_{\hat{k}}^{begin} - x_k^{end} \geq g_{k,\hat{k}}^{connection} \quad \forall k, \hat{k} \in E; k, \hat{k} \in E^{connection} \tag{16}$$

Variables constraints:

$$x_k^{begin}, x_k^{end}, z_k \geq 0 \quad \forall k \in E \tag{17}$$

$$\gamma_{k,\hat{k}}, \lambda_{k,\hat{k}} \in \{0,1\} \quad \forall k, \hat{k} \in L_j, j \in B; k < \hat{k} \tag{18}$$

$$q_{k,j,tc} \in \{0,1\} \quad \forall k \in L_j, tc \in P_j, j \in B \tag{19}$$

$$\varepsilon_k \in \{0,1\} \quad \forall k \in E \tag{20}$$

$$Buff_k \in [0, Buff_{\max}] \quad \forall k \in E \tag{21}$$

A detailed explanation of the above constraints is reported in Appendix A.

The above optimization problem has at most $n_{\text{var}} = 8 \cdot |E|$ variables and $n_{const} = |K_i| \cdot |T| + 9 \cdot |E| + 2 \cdot |L_j| \cdot |B| + 4 \cdot |L_j|^2 \cdot |B| + |K_i \cap L_j| \cdot |P_j| \cdot |B| \cdot |T| + |E^{connection}| + |L_j| \cdot |P_j| \cdot |B|$ constraints.

We remark that the above problem statement, although inspired by [32], has several important differences: first of all, in the first part of the objective function we consider the term (z_k+$Buff_k$·h_k), i.e., the extended delay in case of stop in station events, rather than the single delay z_k; second, we consider an additional term in the objective function that measures the robustness of the timetable to further disturbances; third, we include all the necessary constraints to model robustness.

The above real-time optimization procedure returns the optimal timetable in *TH* and guarantees that no conflict occurs in *TH*. However, given the dimension of the problem, the requirement to obtain a rescheduled timetable in a short computation time imposes the time horizon *TH* to be limited and in general shorter than the time window *TW* in which the overall timetable has to be defined (which may typically last up to 24 h). Therefore, the following Step 2.1 is applied to extend the rescheduling to the whole time window.

2.2. Step 2.1: the real-time merging procedure

The merging procedure can be easily described as follows: all the events not included in *TH* but present in the nominal timetable and necessary for each rescheduled train to reach its final destination are shifted according to the optimal timetable in *TH*. All remaining events keep their nominal scheduling. The extended timetable in *TW* ends when the last train trip affected by the initial disturbance has reached its final destination. As previously discussed, the optimal timetable in *TH* is conflict-free within the time horizon. On the contrary, the merging procedure can lead to new conflicts arising after *TH* because of the presence of shifted events. The heuristic procedure in Step 2.2 allows identifying and solving such possible new conflicts.

2.3. Step 2.2: the real-time heuristic procedure

The real-time heuristic procedure is summarized in Fig. 3. This heuristics mimics a job-shop scheduling problem [37], which aims at minimizing secondary delays while identifying and solving conflicts. In simple words, given the extended timetable in *TW*, the procedure detects the first conflict eventually arising after the time horizon, solves it, and goes on iteratively identifying and solving all conflicts until the last perturbed train has reached its final destination. More in detail, first, unfeasible coincidences in stations are removed (*PHASE1*). To this aim, all time values associated with each train (i.e., its arrival and departure time at each station) are checked: if a coinciding value between two trains is found, a waiting time is assigned to the train with a lower priority in order to maximize the service quality. Priority is given to direct trains or to trains with the highest traveling time. Note that different priorities may also be adopted according to the railway company policy, for instance favouring trains that on average transport more passengers. The waiting time in case of coincidences is assigned according to the following rule: to avoid simultaneous arrivals (in case of trains travelling in the same direction) and/or arrival-departure (in case of trains travelling in opposite directions) of two trains in a station, the train with lower priority has to wait in its previous station for the minimum waiting time (that is, the time required to allow passengers to get on and off from the other train, as established by the nominal timetable). Subsequently, both trains are allowed to depart, and, in case they are traveling in opposite directions and a crossing on a single-track line is thus generated, this will be solved in the next phase of the heuristics.

Once all unfeasible coincidences are solved, the first eventual crossing in a single-track line segment is identified (*PHASE2*). In this case the train with lower priority has to wait in the station located before the crossing point until the arrival of the highest priority train plus a safety time (as established by norms).

The proposed heuristics is iterative, so that, in case of new unfeasible coincidences, they are solved going back to *PHASE1*. The procedure ends when the last perturbed train has reached its final destination and all conflicts are solved. Therefore, at the end of Step 2.2, the TD obtains the rescheduled timetable in *TW* and can restore the proper functioning of the network.

2.4. Step 3: the offline self-learning procedure

The proposed decision making procedure requires as input of Step 1 the relevance weights (α,β) to be used in Eq. (1) that are stored in the database. The goal of Step 3 is that of identifying offline the most suitable weights, according to some conflicting criteria and based on the real duration of the disturbance. In such a way, the new weights are stored in the database and fruitfully used if disturbances of the same type occur in the future.

The proposed procedure is based on the concept of fuzzy cross-efficiency Data Envelopment Analysis [11]. It consists in a generalization of the traditional DEA, whose employment for train rescheduling has been proposed up to now only in

```
Algorithm: Heuristic procedure in Step 2.2
  Input: ET = Extended Timetable in TW;
Set RT = Rescheduled Timetable in TW = ET;
run PHASE1
for (all couples of Trains (i_a, i_b) in RT)
    while (unfeasible coincidence) == true
        if (Travelling Time(i_a) > Travelling Time(i_b) OR i_a is direct)
            stop i_b in the previous station;
            departure(i_b) = departure(i_b) + waiting time;
        else
            stop i_a in the previous station;
            departure(i_a) = departure(i_a) + waiting time;
run PHASE2
for (all couples of Trains (i_a, i_b) in RT)
    while (crossing at a single-track line) = true
        if (Travelling Time(i_a) > Travelling Time(i_b) OR i_a is direct)
            stop i_b in the previous station;
            departure(i_b) = arrival(i_a) + safety time;
        else
            stop i_a in the station;
            departure(i_a) = arrival(i_b) + safety time;
for (all couples of Trains (i_a, i_b) in RT)
        if (unfeasible coincidence) = true
            go back to PHASE1
        else
            update RT
Output RT.
```

Fig. 3. The pseudo-code summarizing the heuristic procedure in Step 2.2.

[34] but in a deterministic setting. Since the decision making procedure is based on statistical evaluations and data affected by imprecisions and uncertainty, we choose the fuzzy DEA approach which is more appropriate in such context. In particular, it allows effectively representing by means of fuzzy numbers the uncertainty affecting performance indices. Technical details on this approach are summarized in Appendix B. In the sequel we show how the fuzzy cross-efficiency DEA technique is used to define the offline self-learning procedure.

A series of F couples of relevance weights (α,β) is generated. For each of them, the optimization procedure of Step 1 is executed considering the real duration of the occurred disturbance. Thus, different timetables are provided, and their effectiveness is evaluated on the basis of some conflicting criteria: the delay at each station (to be minimized) and some robustness indices (to be maximized). The fuzzy cross-efficiency DEA allows us to select the most efficient couple of weights according to such conflicting criteria, as described in Appendix B.

The first robustness index to be maximized is R defined in Eq. (2) and is a measure of the robustness of the overall timetable. The other robustness indices are the *Weighted Average Distance* (*WAD*) calculated for each train $i \in T$ [24]:

$$WAD_i = \frac{\sum_{k \in K_i} \frac{(2 \cdot k - 1) \cdot Buff_k \cdot h_k}{2 \cdot N_i^{st}}}{\sum_{k \in K_i} Buff_k \cdot h_k}, i \in T, \tag{22}$$

where N_i^{st} is the number of stations encountered by the ith train along its trip within the time horizon. In simple words, index WAD_i describes the buffer time distribution along the trip for the ith train and can assume values between 0 and 1. For example, a value of $WAD_i = 0.5$ corresponds to the fact that on average an equal amount of buffer times is allocated in the first half and in the second one of the train trip, while values smaller (bigger) than 0.5 relate to a shift in the buffer times distribution towards the beginning (end) of the train trip. Usually, it is preferable to have time reserves concentrated

Table 2
Infrastructural data of the railway network.

Segm.	Description	Type	Nr. of tracks
j_0	Mungivacca	Station	2
j_1	Mungivacca–Triggiano	Single track	1
j_2	Triggiano	Station	2
j_3	Triggiano–Capurso	Single track	1
j_4	Capurso	Station	2
j_5	Capurso–Noicattaro	Single track	1
j_6	Noicattaro	Station	2
j_7	Noicattaro–Rutigliano	Double track	2
j_8	Rutigliano	Station	2
j_9	Rutigliano–Conversano	Single track	1
j_{10}	Conversano	Station	2
j_{11}	Conversano–Castellana	Single track	1
j_{12}	Castellana	Station	2
j_{13}	Castellana–Grotte di Castellana	Single track	1
j_{14}	Grotte di Castellana	Unitary station	1
j_{15}	Grotte di Castellana–Putignano	Single track	1
j_{16}	Putignano	Station	2

early on the line (i.e., a small WAD_i value) [26]. However, if disturbances occur later on the line, the time reserves located previously to the occurrence may be of no use. Hence, authors in [38] state that the average amount of time reserves should be allocated on the middle part of a line, with a slight shift to the beginning.

Using the fuzzy cross-efficiency DEA technique, a *Defuzzified Cross-Efficiency* (DCE_f) value is computed for each couple of weights considered ($f=1,\dots,F$). Based on such values, the F alternatives are ranked and the most appropriate couple of relevance weights is determined.

Problems in Step 3 (see Appendix B) have a number of variables equal to $3\cdot(W+H)\cdot F$ and a number of constraints equal to $3\cdot(2\cdot F+W+H)$.

After performing Step 3, the database of the decision making procedure is updated with the relevance weights related to the most efficient alternative and with the type and duration of the occurred disturbance, so that, the next time a disturbance of the same type affects the railway traffic, Step 1 of the procedure receives in input more appropriate weights.

3. A real case study

Before introducing the case study, we point out that for sake of clarity in this section we adopt a notation that slightly differs from the one in Section 2.1. In more detail, as regards trains, segments, and tracks, we use the respective identifying number as subscript of the corresponding symbol, instead of reporting the number assignment. For example, when referring to the first train, we write i_1 instead of $i=1$, similarly the first segment is identified as j_1, and so on.

3.1. The considered railway network

In this section we illustrate and test the proposed approach on a real case study. To describe it in detail and even allow replication, we voluntarily consider a railway network with a simple topology, namely a corridor, even if the approach can be applied to any other network. Furthermore, such topology characterized by the presence of single tracks, allows evaluating the effectiveness of the procedure in case of strict constraints. Indeed, in many stations only a train at a time can stop or pass through. As a result, the efficient management of the railway traffic flow in the considered case study requires special attention.

We consider a portion of a regional railway network, namely “Ferrovie del Sud Est” (hereinafter FSE), located in Apulia (Southern Italy), described in Table 2.

It includes the stations between the sites of Mungivacca (i.e., segment j_0) and Putignano (i.e., segment j_{16}), and its traffic is currently managed by a TD using a CTC (Centralized Traffic Control) system, which provides a centralized control for signals and switches within a limited territory, using a single control console. The operative system is installed in j_0 that is an independent station, not controlled by the CTC, as is j_{16}, whereby these stations are not studied here. The line is single-tracked except for segment j_7, which is double-tracked. Moreover, segment j_{14} is a unitary capacity station.

Railway directions are described as even and odd, respectively corresponding to trains going north and south through the network; in the considered network each day there are 24 even trains and 22 odd trains. A safety time $\Delta_j^M = 3$ minutes is assumed for two trains traveling in opposite directions of the same jth segment, and a safety time $\Delta_j^F = 1$ minute for trains in the same direction of the segment. Due to the average duration of the stop in station events, the $Buff_{max}$ parameter is here set to 4 min.

Data on several months have been exploited, showing that the proposed methodology outperforms the current technique used by FSE, which is largely based on the manual rescheduling approach (that is, based on the TD’s personal experience and

Table 3
Results from the real-time optimization procedure (Step 1).

Index	Value
Cumulative delay [min]	242.00
Average delay at stations [min]	7.14
Robustness index R	2.08

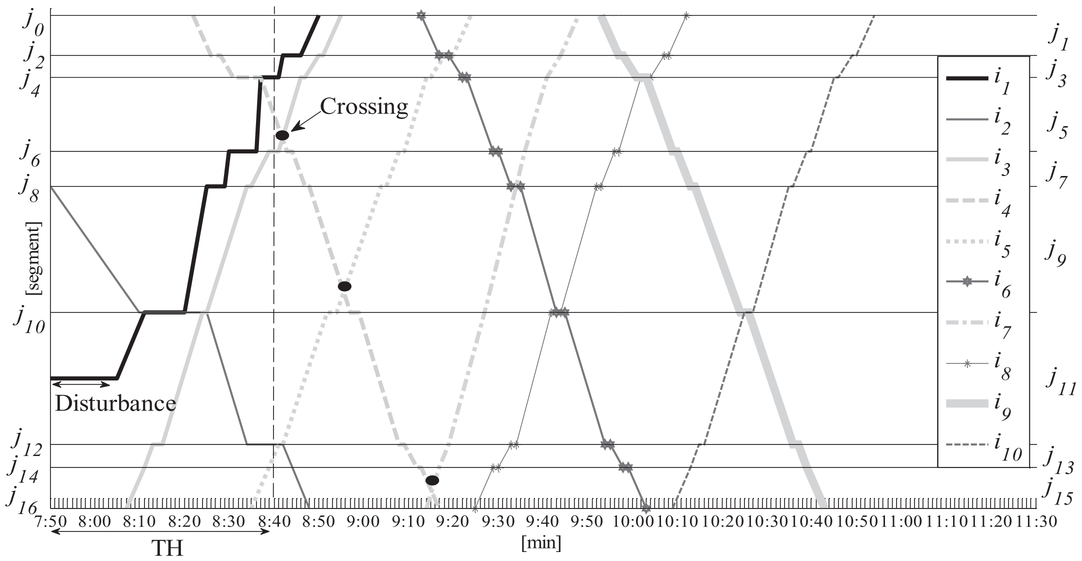

Fig. 4. Extended timetable in *TW* (after Step 2.1).

considering the existing physical and safety constraints). In the next subsection we detail the application of the proposed decision making procedure to a typical scenario among the available data sets. To confirm the effectiveness of the proposed procedure, we validate the approach results firstly by comparing them with those obtained by solving the same case study manually (i.e., by the TD), and then by evaluating the robustness of the solution when a random additional disturbance on the same line is observed.

3.2. Application of the proposed decision making procedure

A real data set is considered referring to a train going from j_{16} to j_0 that stops along the line at segment j_{11} for the occurrence of a disturbance, which consists in a track unavailability (e.g., for the presence of an obstacle on the line) and occurring at 7:50 am, a rush hour with high level of traffic. Based on historical data, the estimated duration of such a disturbance is 15 min, and, according to the network topology, the most suitable time horizon is $TH = 50$ minutes, as confirmed by some preliminary assessments that we have carried out (not reported here for the sake of brevity). Therefore, the considered disturbance affects five trains in *TH*. Finally, based on the estimated order of magnitude of the two terms in Eq. (1), the relevance weights assumed for the real-time optimization procedure are initialized at: $\alpha = 1$ and $\beta = 100$.

The MILP problem is solved by the GLPK tool in the MATLAB environment since they are largely used for the resolution of such problems; clearly, other optimizers (such as, for instance, CPLEX) can be successfully used to this aim. The results obtained by applying Step 1 of the proposed procedure are reported in Table 3, showing the cumulative delay of the network (i.e., the first term in Eq. (1)), the average delay at stations, and the overall robustness (i.e., the second term in Eq. (1)) in the chosen time horizon.

By applying the real-time merging procedure in Step 2.1, the extended timetable in *TW* is obtained, as shown in Fig. 4 by means of a Cartesian graph that represents the railway schedule in time and space. This diagram shows stop and running events for all the considered trains, reporting the corresponding duration and direction. In particular, the time line is plotted on the *x*-axis, while the railway line (i.e., the space) is plotted on the *y*-axis. The station segments (reported on the left margin of Fig. 4) are represented by lines parallel to the *x*-axis, while line segments (reported on the right margin of Fig. 4) are the portion of the graph comprised between two consecutive stations. Stop events are represented as segments on station lines, whose length corresponds to the duration of the event. Running events are represented by oblique lines whose orientation indicates the train direction and whose slope represents the corresponding speed. For instance, the running events are oriented from bottom to up for even trains that travel from South to North (i.e., to the station on the upper

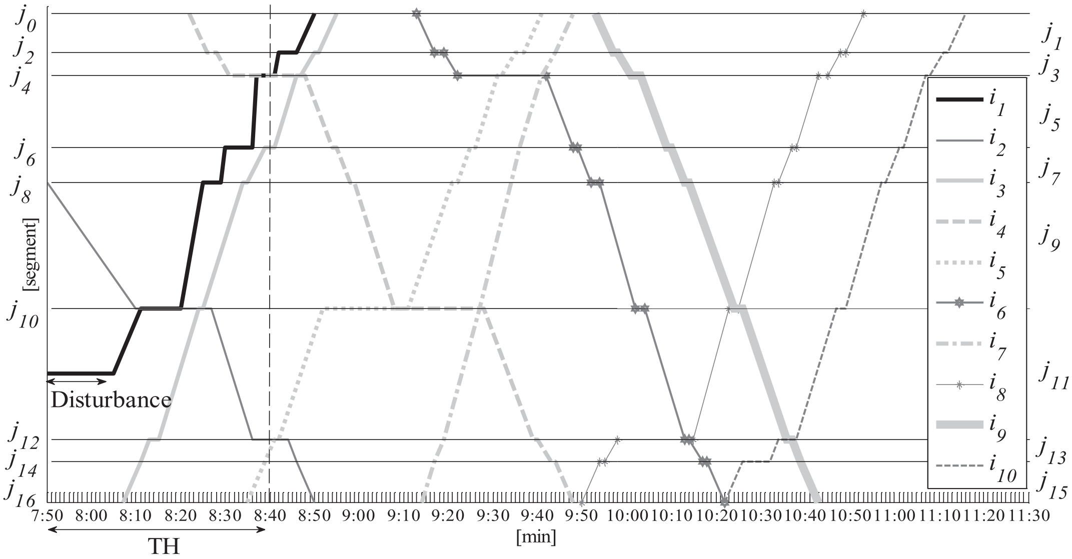

Fig. 5. Rescheduled timetable in *TW* (after Step 2.2).

end of the y-axis, that is, segment j_0). By analyzing Fig. 4, it is worth noting that the application of the optimization procedure allows obtaining a conflict-free timetable within the chosen *TH*. However, applying the merging procedure in Step 2.1, train conflicts may happen at time instants exceeding the considered time horizon. In fact, Fig. 4 shows that trains i_3 and i_4 would cross at segment j_5, i_4 and i_5 at j_9, i_4 and i_7 at j_{15}. All these crossings are at single-track lines, and hence they are unfeasible. Therefore, in order to identify and solve all conflicts arising after the chosen time horizon (up until a total of 24 h), the heuristic procedure in Step 2.2 is applied, calculating a near-optimal solution, so that the TD can choose and communicate the new status to drivers as soon as possible. The corresponding rescheduled timetable in *TW* obtained after the heuristic procedure is graphically reported in Fig. 5. We remark that all conflicts on single tracks are solved. We also highlight that, at the end of Step 2.2, the TD can restore the proper functioning of the network by applying the rescheduled timetable in TW. In the evaluated case study this is possible in about 1 min from the disturbance occurrence.

The offline self-learning procedure in Step 3 is applied to allow a performance evaluation of the obtained solution so as to properly update the database in the case that a similar disturbance occurs in the future. Table 4 reports the considered fuzzy inputs for Step 3. In particular, Table 4a shows the parameters to be minimized (i.e., the delay at each station in *TH*), while Table 4b reports those to be maximized (i.e., the index WAD_i for trains involved in *TH* and the overall robustness *R*). Such values are reported for different possible solutions obtained by varying the relevance weights in the objective function of the optimization procedure in Step 1 and taking into account the real duration of the occurred disturbance. According to the self-learning procedure presented in Section 2.4 and detailed in Appendix B, each parameter reports the corresponding optimistic (*o*), modal (*m*), and pessimistic (*p*) value, where optimistic and pessimistic values are obtained from the modal one (resulting from Step 1) considering the standard deviation of real data and assuming that the pessimistic value is closer to the modal one with respect to the optimistic value (the reader is referred to Appendix B for details on the fuzzy triples).

Note that a WAD_i value is not assigned to trains i_4 and i_5. This is due to the two conflicting goals of the optimization procedure. In fact, on the one hand, the robustness index WAD_i for trains i_4 and i_5 is equal to zero, since, in the considered time horizon, there are no subsequent trains that can be perturbed (i.e., the $NSucT_k$ parameter in Eq. (2) is equal to zero). On the other hand, the delays minimization leads the MILP problem to set buffer times equal to zero (i.e., their lower bound). Hence, WAD_i cannot be determined for $i = 4{,}5$. Also note that in Table 4 we only consider six different alternatives. This is due to the fact that, for small variations of the relevance weights, there are few relevant changes in the examined parameters. Hence, for the sake of brevity, we prefer to report in Table 4 only few rescheduling alternatives, choosing among those that are significantly different from each other.

Table 5 shows the percentage value of the obtained defuzzified cross-efficiencies DCE_f and the final ranking for the considered alternative solutions. By analyzing Table 5 it is important to note that the self-learning procedure confirms that the considered relevance weights (that is, the alternative numbered as #2) represent the best compromise between the two conflicting objectives in Step 1 (i.e., the minimization of all delays and the increase in the robustness). In fact, alternative #2 has the highest efficiency value. It is also worth noting that the obtained results are consistent with the order of magnitude of the two terms in Eq. (1) (see Table 3). By increasing the ratio between the two relevance weights, it can be observed that the efficiency value decreases; in fact, although the robustness value increases, at the same time there is a greater rise in delays at the stations, thus leading to lower efficiency values. On the contrary, alternative #1 (i.e., with $\alpha = 1$ and $\beta = 1$) has the worst efficiency; in fact, with such values of the relevance weights the first term of the objective function (1) is too predominant over the second one, and thus the obtained timetable robustness is very limited.

Table 4
Data for Step 3 of the decision making procedure: parameters to be minimized (a) and maximized (b).

(a) *Parameters to be minimized*

Altern.# ($\alpha;\beta$)		z_2[min]	z_4 [min]	z_6 [min]	z_8 [min]	z_{10} [min]	z_{13} [min]	z_{15} [min]	z_{22} [min]	z_{24} [min]	z_{26} [min]	z_{28} [min]
1 (1;1)	*o*	11.70	14.40	13.50	13.50	12.60	0.00	0.00	9.00	9.00	9.00	8.10
	m	13.00	16.00	15.00	15.00	14.00	0.00	0.00	10.00	10.00	10.00	9.00
	p	14.95	18.40	17.25	17.25	16.10	0.00	0.00	11.50	11.50	11.50	10.35
2 (1;100)	*o*	11.70	14.40	13.50	13.50	12.60	9.90	13.50	0.90	0.00	0.00	0.00
	m	13.00	16.00	15.00	15.00	14.00	11.00	15.00	1.00	0.00	0.00	0.00
	p	14.95	18.40	17.25	17.25	16.10	12.65	17.25	1.15	0.00	0.00	0.00
3 (1;500)	*o*	11.70	18.00	21.60	21.60	20.70	9.90	13.50	0.90	3.60	3.60	2.70
	m	13.00	20.00	24.00	24.00	23.00	11.00	15.00	1.00	4.00	4.00	3.00
	p	14.95	23.00	27.60	27.60	26.45	12.65	17.25	1.15	4.60	4.60	3.45
4 (1;1000)	*o*	11.70	18.00	21.60	25.20	24.30	12.60	13.50	3.60	6.30	6.30	5.40
	m	13.00	20.00	24.00	28.00	27.00	14.00	15.00	4.00	7.00	7.00	6.00
	p	14.95	23.00	27.60	32.20	31.05	16.10	17.25	4.60	8.05	8.05	6.90
5 (1;2000)	*o*	11.70	18.00	21.60	25.20	24.30	12.60	13.50	3.60	6.30	7.20	6.30
	m	13.00	20.00	24.00	28.00	27.00	14.00	15.00	4.00	7.00	8.00	7.00
	p	14.95	23.00	27.60	32.20	31.05	16.10	17.25	4.60	8.05	9.20	8.05
6 (1;3000)	*o*	11.70	18.00	21.60	25.20	24.30	12.60	13.50	3.60	6.30	9.90	9.00
	m	13.00	20.00	24.00	28.00	27.00	14.00	15.00	4.00	7.00	11.00	10.00
	p	14.95	23.00	27.60	32.20	31.05	16.10	17.25	4.60	8.05	12.65	11.50

(b) *Parameters to be maximized*

Altern.# ($\alpha;\beta$)		WAD_1 [%]	WAD_2 [%]	WAD_3 [%]	R
1 (1;1)	*o*	0.12	0.00	0.14	1.65
	m	0.10	0.00	0.13	1.44
	p	0.09	0.00	0.11	1.30
2 (1;100)	*o*	0.12	0.29	0.14	2.41
	m	0.10	0.25	0.13	2.09
	p	0.09	0.23	0.11	1.88
3 (1;500)	*o*	0.35	0.29	0.37	2.80
	m	0.30	0.25	0.33	2.44
	p	0.27	0.23	0.29	2.19
4 (1;1000)	*o*	0.46	0.29	0.29	2.87
	m	0.40	0.25	0.25	2.49
	p	0.36	0.23	0.23	2.24
5 (1;2000)	*o*	0.46	0.29	0.29	2.87
	m	0.40	0.25	0.25	2.49
	p	0.36	0.23	0.23	2.24
6 (1;3000)	*o*	0.46	0.29	0.43	2.87
	m	0.40	0.25	0.38	2.502
	p	0.36	0.23	0.34	.25

Table 5
Results of Step 3 of the proposed decision making procedure.

Alternative # ($\alpha;\beta$)	DCE_f [%]	Ranking
1 (1;1)	6.65	6
2 (1;100)	68.01	1
3 (1;500)	49.57	2
4 (1;1000)	37.94	3
5 (1;2000)	33.23	4
6 (1;3000)	24.21	5

Finally, Table 6 summarizes the computation times of the different phases of the decision making procedure when using a 3.40 GHz processor and 8 GB RAM PC. As it can be seen, the most burdensome part is Step 2.2 that requires almost 60 s. This is definitely consistent with an application in real time.

3.3. Validation of the proposed robust real-time rescheduling

By analyzing the data in Table 4a, it can be observed that the initial delay is absorbed without spreading over the network (especially in alternative #2), thus providing a first confirmation of the fact that the obtained timetable is robust according to the robustness characteristic features provided in [21] and [22], despite the actual values of the relevance weights.

Table 6
Problems dimensions and computation times.

Step	Problem dimension	Computation time [s]
Step 1	296 × 326 (variables × constraints)	1.90
Step 2.2	14 (solved conflicts)	59.80
Step 3	198 × 81 (variables × constraints)	11.90

Table 7
Comparison of the decision making procedure with a manual rescheduling.

Index	Step 1 + Step 2	Manual rescheduling	Reduction [%]
Cumulative delay [min]	242.00	1284.00	81.20%
Average delay at stations [min]	7.14	14.00	49.00%

Table 8
Robustness validation with an additional stochastic disturbance in $TH = 50'$.

Duration of additional disturbance [min]	Percentage of conflict-free timetables [%]			Average delay at stations within *TH* [min]		
	Delays min. + robustness max. (%)	Delays minim.(%)	Saving [%]	Delays min. + robustness max.	Delays minim.	Saving [%]
1.0	77.00	31.00	148.40	0.04	0.19	79.00
1.5	100.00	50.00	100.00	0.00	0.14	100.00
2.0	73.00	27.00	170.40	0.03	0.25	88.00
2.5	46.15	53.84	−14.30	0.14	0.20	30.00
3.0	18.18	18.18	0.00	0.41	0.72	43.10
3.5	46.15	46.15	0.00	0.55	0.82	32.90
4.0	15.38	46.00	−66.60	0.88	0.89	1.10

In the following, to further validate the presented decision making procedure and to evaluate its impact on the company's performance, we firstly compare the obtained results with those obtained by manually solving the same case study already considered. Although there are no specific compulsory rules, usually TDs evaluate if, as a consequence of the occurred disturbance, any crossing or unfeasible coincidences arise, and, if so, they solve them one at a time, by applying almost the same logic already described in the heuristic procedure of Step 2.2. Hence, we compare the results obtained by applying the proposed robust real-time rescheduling (i.e., Step 1 + Step 2.1 + Step 2.2) with those obtained by a manual rescheduling, (i.e., applying the heuristics in Step 2.2 to the whole time window *TW*). Table 7 reports the cumulative delay (first row) and the average delay at stations (second row) resulting from the application of the decision making technique (column 2) and from the application of the manual rescheduling (column 3). It is worth noting that by applying the proposed procedure the cumulative delay is equal to 242 min, while with the manual rescheduling it results equal to 1284 min. Consequently, the presented rescheduling technique ensures a reduction of the cumulative delay for the case study higher than 80%. Moreover, when comparing the average delay at stations, the resulting reduction equals about 50%. It is also to be noticed that, by performing a manual rescheduling, errors or larger delays may arise, and the effectiveness of the adopted control actions cannot be properly evaluated.

Furthermore, in order to validate the robustness of the obtained solution, we compare the optimal timetable in *TH* (that is, the timetable obtained by applying Step 1 of the proposed decision making procedure) with that obtained by only minimizing the cumulative delay (i.e., by solving the rescheduling mathematical problem proposed in [10]). To this aim, we assume that an additional small disturbance (e.g., a lamp failure on a colour light signal) occurs in the network after the initial primary disturbance has been coped with. To test the robustness of the rescheduled timetable under different scenarios, we consider different characteristics of the additional disturbance in terms of duration, affected train, and occurrence time. In particular, we perform 100 replications by randomly generating different duration values of the additional disturbance (in the range $[1, Buff_{max}] = [1,4]$ min), a different train directly affected by such a disturbance, and a different time of occurrence. Therefore, we determine for the above rescheduled timetables the average delay at stations and the number of arising conflicts that are solved by the subsequent heuristics. Table 8 reports for different durations of the additional disturbance the obtained results in terms of percentage of conflict-free timetables and average delay at stations within *TH*, as well as the corresponding percentage savings. By analyzing Table 8 it can be noticed that, when the duration of the additional disturbance is limited (i.e., below 2 min), keeping into account the robustness maximization allows obtaining remarkable savings both in terms of the percentage of conflict-free timetables and of the average delay at stations. On the contrary, when the duration of the new disturbance is greater than 2 min, the obtained results show no savings (or even a worsening) in terms of percentage of timetables without conflicts. Nonetheless, in such cases it is worth noting that the average delay at station is still substantially improved when the rescheduling is devoted to taking into account both the delays minimization and the robustness maximization.

We conclude this section remarking that both railway companies and their passengers can take advantages from the presented decision making procedure. Indeed, in case of disturbances, rail companies need to provide their customers with a quality service, reducing delays or limiting travellers' discomfort. The proposed procedure trades off between two conflicting objectives. From the one hand it takes into account the minimization of the delay associated with each intermediate event, instead of the overall delay accumulated by the single train at the end of its route, thus providing lower intermediate passengers' waiting times, while limiting the waiting time over the entire journey. On the other hand, thanks to its ability to provide in real-time a robust rescheduled timetable, the method also allows reducing the spreading over the network of secondary delays and eventual additional primary delays. Moreover, the TD is provided with an automatic support tool that allows to rapidly restore the proper functioning of the network.

Finally, thanks to the self-learning procedure, the quality of the rescheduling is improved at each reapplication of the method. This way, companies can avoid sanctions (i.e., ticket refunds or penalties established by contract for late running), and customers can benefit from a reliable service without loss of money and time. Moreover, an efficient service can produce, as a secondary effect, a substantial shift of passengers (and freight) from other transportation modes to rail, thus helping reducing road traffic congestion and environmental impact.

4. Conclusions and future developments

In this paper we propose a self-learning decision making procedure for real-time rescheduling of trains in mixed-tracked railway networks with acyclic timetable under disturbances. The presented technique firstly applies a mixed integer linear programming optimization in a suitable time horizon to trade-off between the minimization of delays and the maximization of the robustness of the rescheduled timetable. Since the time horizon has to be limited due to real-time requirements, a merging procedure is then applied to extend the rescheduling to a wider time window in which there are still trains affected by the initial disturbance. Subsequently, since the obtained timetable cannot be guaranteed to be conflict-free after the time horizon, a heuristic procedure identifies and solves all conflicts that may arise in the timetable in an extended time interval, thus allowing the train dispatcher restoring in real-time the proper functioning of the railway network. Finally, an offline self-learning procedure allows a performance evaluation of the obtained solution in order to properly update the database in case of future disturbances of the same type. To this aim, the decision making procedure applies a cross-efficiency fuzzy Data Envelopment Analysis to determine the efficiency of the obtained timetable (in terms of minimization of delays at each station and maximization of its robustness) compared to that of other possible solutions and rank the alternatives according to their efficiency values.

The presented procedure is tested and validated on a real case study, demonstrating its effectiveness. The technique is useful for both railway companies (to provide their customers with on-time services, reduce sanctions or penalties, and avoid possible errors caused by a manual rescheduling) and passengers (to reduce waiting times and delays or to limit travel discomforts).

Future work regards validating the proposed decision-making procedure on more complex networks, characterized by heavy traffic conditions and comparing it with other more mature solutions. Moreover, we plan to extend the procedure to address the case of multiple simultaneous disturbances.

Acknowledgment

The authors wish to thank the "FSE - Ferrovie del Sud Est e Servizi Automobilistici" staff for their support in the preparation of this paper.

Appendix A. . Explanation of constraints (3)–(21) in the optimization procedure in Step 1

➢ *Train constraints:*

– Given the ordered set of events K_i associated with train i, whose last element is n_i, constraint (3) forces event $(k+1)$ to begin as soon as event k ends.
– Constraint (4) states that the duration of the rescheduled event k, increased of the corresponding buffer time $Buff_k$, has to be higher than, or equal to, the offline scheduled duration d_k (both for trips, i.e., when $h_k = 0$, and stops, i.e., for $h_k = 1$) when it is lower than the minimum offline scheduled duration (i.e., $d_{k_{trip}}$ for trips or $d_{k_{stop}}$ for stops). However, when the offline scheduled duration is equal to, or higher than, the minimum offline scheduled duration, it is decreased by a factor $\varepsilon_k \cdot \delta_k$ (where the binary variable ε_k is equal to 1 if the extended delay of the considered event k is higher than a fixed threshold w_k, or 0 otherwise). The constant δ_k is a recovery time established in the nominal timetable.
– Constraint (5) imposes that the beginning of the rescheduled event k has to be higher than or equal to the beginning chosen offline in the nominal timetable (i.e., b_k^{nominal}), when the event is a stop in station event (i.e., if $h_k = 1$).
– Constraint (6) states that the beginning of event k has to be equal to its static value (b_k^{static}) if the static value is higher than zero, that is, when event k has started before the occurrence of the disturbance. Similarly, constraint (7) states that the end of event k has to be equal to its static value (e_k^{static}) if the static value is higher than zero, that is, when event k has started before the occurrence of the disturbance.

- Constraint (8) imposes that the difference between the end of the rescheduled event k delayed of the corresponding buffer time $Buff_k$ and its offline scheduled end (as established in the nominal timetable, i.e., e_k^{nominal}) has to be lower than or equal to the corresponding delay z_k.

➢ *Technical constraints:*

- Constraint (9) imposes that a single track cannot be occupied by more than one train at the same time.
- In constraint (10), given two events k and $\hat{k}$ related to a track tc, when k occurs before $\hat{k}$ (as in the offline scheduling) track tc must be reserved to event k, while when k occurs after $\hat{k}$ (as may happen in case of rescheduling), track tc must be reserved to event $\hat{k}$ (that is, either $\gamma_{k,\hat{k}}$ is equal to 1 or $\lambda_{k,\hat{k}}$ is equal to 1).
- In constraint (11a), in case of trains travelling in opposite direction (i.e., when $o_{\hat{k}} \neq o_k$), any event $\hat{k}$ subsequent to event k and requiring the same track used by k, has to start when a Δ_j^M time interval has elapsed after the end of event k. Similarly, in constraint (11b), in case of subsequent trains (i.e., if $o_{\hat{k}} = o_k$), any event $\hat{k}$ subsequent to k and requiring the same track used by k has to start only when a Δ_j^F time interval has elapsed after the end of k.
- In constraint (12a), in case of trains travelling in opposite direction (i.e., $o_{\hat{k}} \neq o_k$) any event k subsequent to $\hat{k}$ and requiring the same track used by $\hat{k}$ has to start when a Δ_j^M time interval has elapsed after the end of $\hat{k}$. Similarly, in constraint (12b), in case of subsequent trains (i.e., $o_{\hat{k}} = o_k$), any event k subsequent to $\hat{k}$ and requiring the same track used by $\hat{k}$ has to start when a Δ_j^F time interval has elapsed after the end of$\hat{k}$.
- Constraint (13) imposes that an event k can either occur after or before a generic event $\hat{k}$.
- Constraint (14) imposes that the length of a generic train i should not exceed the length of the track it occupies.

➢ *Operator preferences:*

- Constraint (15) imposes that, when an event k is rescheduled, the sum of its buffer time $Buff_k$ and of its delay z_k (i.e., the extended delay of the event k) minus a threshold w_k is lower than or equal to a large positive constant M.
- According to constraint (16), event $\hat{k}$ cannot start if k is not ended and a constant time of connection $g_{k,\hat{k}}^{connection}$ between the two events has not elapsed.

➢ *Variables constraints:*

- Constraint (17) states that the beginning and the end of a generic event k, as well as the corresponding delay z_k, are non-negative variables.
- Constraint (18) imposes that $\gamma_{k,\hat{k}}$, i.e., the variable used to specify if event k occurs before $\hat{k}$ (value 1) or not (value 0), and $\lambda_{k,\hat{k}}$, i.e. the variable used to specify if k is rescheduled to occur after $\hat{k}$ (value 1) or not (value 0), are binary variables.
- Constraint (19) imposes that $q_{k,j,\,tc}$, i.e., the variable used to specify if the event k uses track tc of segment j (value 1) or not (value 0), is a binary variable.
- Constraint (20) imposes that ε_k, i.e., the variable used to specify if the delay of event k is higher than a fixed threshold w_k (value 1) or not (value 0), is a binary variable.
- Constraint (21) states that $Buff_k$ can assume real values ranging between 0 and $Buff_{max}$ minutes, where the upper bound is suitably chosen depending on the average duration of the stop in station events.

Appendix B. . Background on the fuzzy cross-efficiency DEA technique adopted in the self-learning procedure in Step 3

Thanks to its robustness and simplicity of application, the Data Envelopment Analysis (DEA) approach is a technique commonly adopted to compare and rank a set of alternatives with heterogeneous operating characteristics under multiple conflicting criteria [11].

DEA allows to compare a set of F alternatives with respect to n conflicting criteria divided into a subset of W criteria to be maximized and a subset of H criteria to be minimized. Criteria are quantified via appropriate performance indices. Now, let $y_{w,f}$ be the value of the wth performance index to be maximized ($w = 1,\ldots,W$) and $x_{h,f}$ be the value of the hth performance index to be minimized ($h = 1,\ldots,H$), both evaluated when the fth alternative is active ($f = 1,\ldots,F$). Then, let $u_{w,f}$ and $v_{h,f}$ be the weighting coefficients associated, respectively, with the wth performance index to be maximized and the hth performance index to be minimized. Such weights, as explained in the following, are used to evaluate the efficiency of the fth alternative.

The efficiency of the fth alternative is defined as the ratio between the weighted sum of the values of the performance indices to be maximized and the weighted sum of the values of the performance indices to be minimized:

$$E_f = \left(\sum_{w=1}^{W} u_{w,f} \cdot y_{w,f} / \sum_{h=1}^{H} v_{h,f} \cdot x_{h,f} \right) \tag{A.1}$$

where the non-negative weighting coefficients result from the solution of the following linear optimization problem:

$$\max \sum_{w=1}^{W} u_{w,f} \cdot y_{w,f} \tag{A.2}$$

s.t.:

$$\sum_{w=1}^{W} u_{w,f} \cdot y_{w,f} - \sum_{h=1}^{H} v_{h,f} \cdot x_{h,f} \leq 0 \tag{A.3}$$

$$\sum_{h=1}^{H} v_{h,f} \cdot x_{h,f} = 1 \tag{A.4}$$

$$u_{w,f}, v_{h,f} \geq 0, w = 1, \dots, W, h = 1, \dots, H. \tag{A.5}$$

The limitation of the traditional DEA method illustrated above is that it places no constraints, other than positivity, on weights, thus allowing the assessment of the efficiency of an alternative using the set of weights that is most favourable to its own circumstances. The most common approach to overcome such limitation is offered by the so-called cross-evaluation [11], which includes both a self- and a peer-evaluation of the considered alternatives, each of which is not only assessed by its own weights but also by the weights of all the other alternatives. In more detail, according to the cross-efficiency DEA, each alternative is measured via F relative efficiency values (each one with respect to one of the others, including itself), and the resulting cross-efficiency of the alternative is the mean value of these F relative efficiencies. Formalizing, a cross-efficiency matrix $CE = \{E_{f_0,f}\} \in (\Re^+)^{FxF}$ is determined, whose generic element $E_{f_0,f}$ represents the efficiency of the f_0th alternative calculated with the most favourable weights of the fth competing alternative (obtained solving optimization problem (A2)–(A5)), namely:

$$E_{f_0,f} = \left(\sum_{w=1}^{W} u_{w,f} \cdot y_{w,f_0} / \sum_{h=1}^{H} v_{h,f} \cdot x_{h,f_0} \right) \tag{A.6}$$

and the cross efficiency of the fth alternative is obtained as:

$$CE_f = \frac{1}{F} \sum_{f_0=1}^{F} E_{f_0,f}. \tag{A.7}$$

The above approach still contains a limitation: indeed, the solution of the linear optimization problem (A2)–(A5) is not unique in general. This has an impact on the values of $E_{f_0,f}$, $f_0 \neq f$. To solve this issue, a second-level procedure is executed for all the f–th alternatives after solving the original problem (A2)–(A5). It consists in stating and solving an additional optimization problem for each alternative f (f = 1,...F), which allows determining a new set of weighting coefficients $u_{w,f}$, $v_{h,f}$ (w = 1,...,W, h = 1,...,H) to be used in Eq. (A6) to redefine the cross-efficiency matrix and finally the cross-efficiency of the alternative. The second level optimization problem is the following:

$$\max \sum_{w=1}^{W} u_{w,f} \left(\sum_{f_0=1, f_0 \neq f}^{F} y_{w,f_0} \right) \tag{A.8}$$

s.t.:

$$\sum_{h=1}^{H} v_{h,f} \cdot \left(\sum_{f_0=1, f_0 \neq f}^{F} x_{h,f_0} \right) = 1 \tag{A.9}$$

$$\sum_{w=1}^{W} u_{w,f} \cdot y_{w,f} - E_f \sum_{h=1}^{H} v_{h,f} \cdot x_{h,f} = 0 \tag{A.10}$$

$$\sum_{w=1}^{W} u_{w,f} \cdot y_{w,f_0} - \sum_{h=1}^{H} v_{h,f} \cdot x_{h,f_0} \leq 0, f_0 = 1, \dots, F (f_0 \neq f) \tag{A.11}$$

$$u_{w,f}, v_{h,f} \geq 0, w = 1, \dots, W, h = 1, \dots, H. \tag{A.12}$$

The above technique allows obtaining the deterministic or crisp cross-efficiencies of all the F alternatives which may be used to define a ranking from the most efficient to the least efficient one. However, in case some performance indices of alternatives are affected by some uncertainty, the previous technique does not reveal effective. Hence, by modelling uncertain performance index values by triangular fuzzy numbers [39], a generalization to the previous approach has been introduced in [11], leading to the so-called fuzzy cross-efficiency DEA. In the sequel we briefly describe the main steps of such an approach. Let us first focus on the performance indices to be minimized. Instead of using a single deterministic value $x_{h,f}$ associated with the generic hth criterion and the generic fth alternative, a triple $\tilde{x}_{h,f} = (x^o_{h,f}, x^m_{h,f}, x^p_{h,f}) \in \Re^3$ -i.e., a triangular fuzzy number- is defined, whose entries are, respectively, the most optimistic, the modal, and the most pessimistic estimate of the fth alternative performance index under the hth criterion. Analogously, the triple $\tilde{y}_{w,f} = (y^p_{w,f}, y^m_{w,f}, y^o_{w,f}) \in \Re^3$ is defined, whose entries are, respectively, the most optimistic, the modal, and the most pessimistic estimate of the fth alternative under the wth criterion to be maximized. Both triples represent triangular fuzzy numbers, i.e., they can assume different real values with a degree of possibility in [0,1], according to suitable triangular membership functions defined in [11]. Hence, for each fth alternative we want to determine a cross-efficiency $C\tilde{E}_f$ which is now a fuzzy variable represented by the triple (CE^p_f, CE^m_f, CE^o_f). In particular, following [11], we compute the triple defining $C\tilde{E}_f$ using an approach that is based on three main goals: (1) the maximization of the modal value CE^m_f, (2) the minimization of the distance of the pessimistic value CE^p_f from the modal value CE^m_f, and (3) the maximization of the distance of the optimistic value CE^o_f from the modal value CE^m_f. In more detail, for each alternative f, a *Positive Ideal Solution* (PIS) is defined as the ideal solution with cross-efficiency triple $(CE^{PIS}_{1,f}, CE^{PIS}_{2,f}, CE^{PIS}_{3,f})$ that simultaneously satisfies goals (1)–(3), i.e., such that:

$$F^{PIS}_{1,f} = \max CE^m_f,\ F^{PIS}_{2,f} = \min[CE^m_f - CE^p_f],\ F^{PIS}_{3,f} = \max[CE^o_f - CE^m_f] \tag{A.13}$$

so that the PIS cross-efficiencies are obtained as:

$$\begin{cases} CE^{PIS}_{1,f} = F^{PIS}_{1,f} - F^{PIS}_{2,f} \\ CE^{PIS}_{2,f} = F^{PIS}_{1,f} \\ CE^{PIS}_{3,f} = F^{PIS}_{1,f} + F^{PIS}_{3,f} \end{cases} \tag{A.14}$$

The PIS is determined solving three problems similar to (A8)–(A12), where the objective functions are in turn the three equations in (A13), and constraints are obtained by appropriately modifying (A9)–(A12) to cope with the fuzzy character of the performance indices (we do not report them here for the sake of brevity but refer to [11] for details). Since in practice such a maximally efficient alternative as the PIS can never be obtained, as goals (1)–(3) can never be reached simultaneously by the same weight set, from the PIS of the fth alternative a compromise fuzzy cross-efficiency value (CE^p_f, CE^m_f, CE^o_f) is determined. The corresponding weight set is determined solving a fuzzy multi-objective linear programming problem by introducing an auxiliary variable and by using a procedure proposed in [40] for a generic fuzzy multi-objective programming problem and adapted to the fuzzy cross-efficiency DEA case in [11].

Finally, the crisp cross efficiencies DCE_f for each alternative $f = 1,\ldots,F$ are determined by defuzzifying the obtained triple using the well-known center of the area method [39]:

$$DCE_f = \frac{CE^p_f + CE^m_f + CE^o_f}{3}. \tag{A.15}$$

The resulting crisp values obtained for each alternative may then be used to define a ranking among them.

References

[1] E. Castillo, I. Gallego, J.M. Urena, J.M. Coronado, Timetabling optimization of a mixed double- and single-tracked railway network, Appl. Math. Model. 35 (2011) 859–878.

[2] M. Dotoli, N. Epicoco, M. Falagario, A. Piconese, F. Sciancalepore, B. Turchiano, A real-time traffic management model for regional railway network under disturbances, in: Proceedings of the Ninth International Conference on Automation Science and Engineering, 2013, pp. 892–897.

[3] X. Guo, J. Wu, H. Sun, R. Liu, Z. Gao, Timetable coordination of first trains in urban railway network: a case study of Beijing, Appl. Math. Model. 40 (2016) 8048–8066.

[4] E. Hassannayebi, S.H. Zegordi, M. Yaghini, Train timetabling for an urban rail transit line using a Lagrangian relaxation approach, Appl. Math. Model. 40 (2016) 9892–9913.

[5] S. Van Aken, N. Bešinović, R.M. Goverde, Designing alternative railway timetables under infrastructure maintenance possessions, Transport. Res. B Methodol. 98 (2017) 224–238.

[6] X. Li, B. Shou, D. Ralescu, Train rescheduling with stochastic recovery time: a new track-backup approach, IEEE Trans. Syst. Man Cybern. Syst. 44 (9) (2014) 1216–1233.

[7] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, J. Wagenaar, An Overview of Recovery Models and Algorithms For Real-Time Railway Rescheduling, Econometric Institute, 2013 Report EI2013-29.

[8] E. Hassannayebi, A. Sajedinejad, S. Mardani, Disruption management in urban rail transit system: a simulation based optimization approach, Handbook of Research on Emerging Innovations in Rail Transportation Engineering, IGI Global, 2016, pp. 420–450.

[9] T. Dollevoet, D. Huisman, L.G. Kroon, L.P. Veelenturf, J.C. Wagenaar, Application of an iterative framework for real-time railway rescheduling, Comput. Oper. Res. 78 (2017) 203–217.

[10] M. Dotoli, N. Epicoco, M. Falagario, B. Turchiano, G. Cavone, A. Convertini, A decision support system for real-time rescheduling of railways, Proc. Eur. Control Conf. (2014) 696–701.

[11] M. Dotoli, N. Epicoco, M. Falagario, F. Sciancalepore, A cross-efficiency fuzzy Data Envelopment Analysis technique for performance evaluation of decision making units under uncertainty, Comput. Ind. Eng. 79 (2015) 103–114.

[12] J. Törnquist, Computer-based decision support for railway traffic scheduling and dispatching: a review of models and algorithms, in: Proceedings of the Fifth Workshop on Algorithm Methods Models for Optimization Railways, 2006.
[13] A. D'Ariano, Innovative decision support system for railway traffic control, Intell. Transp. Syst. Mag. 1 (2009) 8–16.
[14] F. Corman, L. Meng, A review of online dynamic models and algorithms for railway traffic management, IEEE Trans. Intell. Transp. Syst. 16 (3) (2015) 1274–1284.
[15] W. Fang, S. Yang, X. Yao, A survey on problem models and solution approaches to rescheduling in railway networks, IEEE Trans. Intell. Transp. Syst. 16 (6) (2015) 2997–3016.
[16] X. Xu, K. Li, L. Yang, J. Ye., Balanced train timetabling on a single-line railway with optimized velocity, Appl. Math. Model. 38 (2014) 894–909.
[17] Y. Gao, L. Yang, S. Li., Uncertain models on railway transportation planning problem, Appl. Math. Model. 40 (2016) 4921–4934.
[18] E.V. Andersson, A. Peterson, J.K. Törnquist, Quantifying railway timetable robustness in critical points, J. Rail Transp. Plan. Manag. 3 (3) (2013) 95–110.
[19] M.A. Salido, F. Barber, L. Ingolotti, Robustness for a single railway line: analytical and simulation methods, Exp. Syst. Appl. 39 (2012) 13305–13327.
[20] E. Hassannayebi, S.H. Zegordi, M.R. Amin-Naseri, M. Yaghini, Train timetabling at rapid rail transit lines: a robust multi-objective stochastic programming approach, Oper. Res. 17 (2) (2017) 435–477.
[21] T. Dewilde, P. Sels, D. Cattrysse, P. Vansteenwegen, Defining robustness of a railway timetable, in: *Proceedings of the Fourth International Seminar on Railway Operational Modelling and Analysis*,, RailRome, 2011, pp. 1–20.
[22] L.G. Kroon, D. Huisman, G. Maróti, Optimisation models for railway timetabling, in: I. Hansen, J. Pachl (Eds.), Railway Timetable and Traffic, Eurailpress, Hamburg, 2008.
[23] A. D'Ariano, D. Pacciarelli, M. Pranzo, Assessment of flexible timetables in real-time traffic management of a railway bottleneck, Transp. Res. C Emer. 16 (2) (2008) 232–245.
[24] L.G. Kroon, R. Dekker, M.J. Vromans, Cyclic Railway timetabling: A stochastic Optimization Approach, Springer, Berlin Heidelberg, 2007, pp. 41–66.
[25] M.A. Salido, F. Barber, L. Ingolotti, Robustness in railway transportation scheduling, in: Proceedings of the Intelligent Control Automation Conference, 2008, pp. 2880–2885.
[26] M. Fischetti, D. Salvagnin, A. Zanette, Fast approaches to improve the robustness of a railway timetable, Transp. Sci. 43 (3) (2009) 321–335.
[27] C. Liebchen, M. Schachtebeck, A. Schoebel, S. Stiller, A. Prigge, Computing delay resistant railway timetables, Comput. Oper. Res. 37 (5) (2010) 857–868.
[28] R. Larsen, M. Pranzo, A. D'Ariano, F. Corman, D. Pacciarelli, Susceptibility of optimal train schedules to stochastic disturbances of process times, Flex. Serv. Manuf. J. 26 (4) (2014) 466–489.
[29] P. Jovanović, P. Kecman, N. Bojović, D. Mandić, Optimal allocation of buffer times to increase train schedule robustness, Eur. J. Oper. Res. 256 (1) (2017) 44–54.
[30] M. Velasquez, P.T. Hester, An analysis of multi-criteria decision making methods, Int. J. Oper. Res. 10 (2) (2013) 56–66.
[31] R. Markovits-Somogy, Ranking efficient and inefficient decision making units in Data Envelopment Analysis, Int. J. Traffic Transp. Eng. 1 (4) (2011) 245–256.
[32] J. Törnquist, A. Persson, N-tracked railway traffic re-scheduling during disturbances, Transp. Res. B 41 (2007) 342–362.
[33] B. Adenso-Díaz, M.O. González, P. González-Torre, On-line timetable re-scheduling in regional train services, Transp. Res. B 33 (6) (1999) 387–398.
[34] M. Samà, C. Meloni, A. D'Ariano, F. Corman, A multi-criteria decision support methodology for real-time train scheduling, J. Rail Transp. Plan. Manag. 5 (3) (2015) 146–162.
[35] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra, N. Mazzino, D. Anguita, Dynamic delay predictions for large-scale railway networks: deep and shallow extreme learning machines tuned via thresholdout, IEEE Trans. Syst. Man Cybern. Syst. (2017) In press, doi:10.1109/TSMC.2017.2693209.
[36] X. Hu, N. Cui, E. Demeulemeester, L. Bie, Incorporation of activity sensitivity measures into buffer management to manage project schedule risk, Eur. J. Oper. Res. 249 (2) (2016) 717–727.
[37] I. Şahin, Railway traffic control and train scheduling based on inter-train conflict management, Transp. Res. B 33 (1999) 511–534.
[38] M.J. Vromans, R. Dekker, L.G. Kroon, Reliability and heterogeneity of railway services, Eur. J. Oper. Res. 172 (2006) 647–665.
[39] H.-J. Zimmermann, Fuzzy Set Theory and its Applications, fourth ed., Springer Science & Business Media, New York, 2001.
[40] M. Jimenez, A. Bilbao, Pareto-optimal solution in fuzzy multi-objective linear programming, Fuzzy Sets Syst. 160 (2009) 2714–2721.