Delft University of Technology

Delft Center for Systems and Control

Technical report 19-013

Railway Disruption: A Bi-Level Rescheduling Algorithm*

G. Cavone, L. Blenkers, T. van den Boom, M. Dotoli, C. Seatzu, and B. De Schutter

To cite this work, please refer to the published version:

G. Cavone, L. Blenkers, T. van den Boom, M. Dotoli, C. Seatzu, and B. De Schutter, "Railway disruption: A bi-level rescheduling algorithm," *Proceedings of the 6th International Conference on Control, Decision and Information Technologies (CoDIT'19)*, Paris, France, pp. 54–59, Apr. 2019. doi:10.1109/CoDIT.2019.8820380

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

Railway disruption: a bi-level rescheduling algorithm

G. Cavone, *Member, IEEE*, L. Blenkers, T. van den Boom, M. Dotoli, *Senior Member, IEEE*, C. Seatzu, *Senior Member, IEEE*, B. De Schutter, *Fellow, IEEE*

Abstract— **The real-time rescheduling of railway traffic in case of unexpected events is a challenging task. This is mainly due to the complexity of the railway service, which has to ensure safety, punctuality, and efficiency to customers by respecting timetable, framework, and resources constraints. Most of the available researches focus on short delays (i.e., disturbances). Approaches typically rely on simplified macroscopic models for large-scale systems or detailed microscopic models for one or a few lines, due to the long computation time required for solving the rescheduling problem. Only a small number of works considers rescheduling in case of long delays (i.e., disruptions) and all of them are also based on either a macroscopic or a microscopic model. This research focuses on disruptions and aims at filling the gap between macroscopic and microscopic modeling by proposing an innovative bi-level rescheduling algorithm based on a mesoscopic Mixed Integer Linear Programming (MILP) model. The technique allows obtaining a feasible rescheduled timetable in a short computation time respecting not only timetable and safety constraints (typical of macroscopic models) but also capacity and ordering constraints for the disrupted stations (typical of microscopic models). The bi-level algorithm first solves the macroscopic MILP rescheduling problem and then, considering the cancellation and non-admissible platform assignments results, it solves a mesoscopic MILP rescheduling problem. This allows to significantly reduce the search space and consequently the computation time. The method is tested for the rescheduling of the Dutch railway traffic in case of a full blockade between two consecutive stations.**

I. Introduction

Increasing the market share of railway transport is one of the top priorities of many governments for the resolution of mobility problems. A proper rescheduling of the railway traffic in case of unexpected events is then fundamental to improve the performance of railway services. Basically, train rescheduling consists in retiming the offline scheduled traffic (i.e., the nominal timetable) to minimize undesired effects (e.g., train delays, customer discomfort, energy consumption) when unpredictable events occur in the network [1]–[3]. 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 breakdowns of trains or infrastructure) [4]–[6]. Both kinds of events cause the nominal timetable to become invalid because at least one train deviates from its original schedule. Generally, Train Dispatchers (TDs) manage disturbances mostly manually based on their experience and knowledge [5]. This process becomes complex in case of disruptions, due to rolling stock constraints. In these cases TDs use contingency plans to manage traffic. This highly stressful and time-consuming manual approach often leads to suboptimal outcomes, since only a limited number of solutions can be reviewed for a rapid decision-making process. Differently, automated real-time rescheduling procedures can refine and speed up the manual approach, thus supporting TDs in determining in real time suitable control actions and updating timetables while optimizing some traffic performance indices. However, due to the scale, the complexity, and the short resolution time constraints, these problems still remain challenging in the related research field.

This paper proposes a bi-level algorithm for the real-time resolution of the rescheduling problem in case of a full blockade between two consecutive stations (i.e., the traffic is prevented in both directions between the two disrupted stations). The rescheduling problem is set in a Mixed Integer Linear Programming (MILP) fashion aiming at minimizing the delays, the cancellations, and the shunting in station. More in detail, a macroscopic and a mesoscopic constrained model of the disrupted network are presented. The macroscopic model considers a high-level representation of the system and allows reordering of trains on tracks, cancellations of train runs, short-turning, and shunting in station; while the mesoscopic model also includes specific control actions in the disrupted stations, i.e., platform assignment and train ordering on platforms. Then, based on the two models, two MILP problems are set and the bi-level algorithm solves them sequentially. The first-level optimization solves the macroscopic MILP problem for the disrupted network, ideally assuming that the stations involved in the disruption have infinite capacity and no platform constraint is necessary. Then, the second-level optimization solves the mesoscopic MILP problem, which includes the additional capacity constraints keeping into account the results of the first-level optimization regarding the assigned cancellations and the non-admissible short turnings in the form of equality constraints. The application of the proposed algorithm leads to a twofold advantage. On one hand, it strongly reduces the required computation time with respect to the resolution of the fully mesoscopic optimization problem. On the other hand, it allows to take into account the capacity limitations of the disrupted stations that otherwise are neglected in the

G. Cavone and M. Dotoli are with the Department of Electrical and Information Engineering, Polytechnic of Bari, Bari, Italy (corresponding author: graziana.cavone@poliba.it) (email: mariagrazia.dotoli@poliba.it)

L. Blenkers, T. van den Boom, and B. De Schutter are with the Delft Center for Systems and Control, Technology University of Delft, Delft, The Netherlands (email: L.Blenkers@student.tudelft.nl) (email: A.J.J.vandenBoom@tudelft.nl) (email: b.deschutter@tudelft@nl)

C. Seatzu is with the Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy(email: seatzu@diee.unica.it)

macroscopic formulation.

The technique is tested for the rescheduling of the national Dutch railway traffic in case of a full blockade between two consecutive stations.

II. State of the art

The *computer-based* rescheduling of railway traffic can be applied when unpredictable events, such as *disturbances* and *disruptions*, occur in the system. Disruptions can be roughly divided into *full blockades* (i.e., no traffic is allowed in a track section) and *partial blockades* (i.e., the capacity of the track section is decreased). In this paper, the focus is on full blockades since they are rarer but more difficult to manage, with respect to disturbances and partial blockade, and induce severe limitations to the railway traffic.

Three main classes of computer-based rescheduling approaches to railway traffic control can be identified in the literature [7]: simulation-based approaches, heuristic procedures, and mathematical optimization. Simulation-based approaches aim at reproducing the flow of real life in which there is no specific objective function [8]. Heuristic procedures take decisions that aim at ensuring a proper behavior of the system, are able to avoid conflicts, give priorities, and so on (see, e.g., [9]). Usually, the computational effort in implementing heuristics is low. Finally, mathematical optimization models have instead a well-defined objective function, which frequently refers to average or maximum delays [10], total delays (i.e., considering the delay at the final destination of trains) [11] or delays at stations along the train trip.

An exhaustive discussion on commonly used rescheduling mathematical models can be found in [12], showing that rescheduling techniques based on mathematical models can provide optimal solutions. However, their implementation and resolution is not trivial, especially when the number of variables and constraints of the problem is high and the rescheduling time horizon is large. Nonetheless, effective solvers (e.g., CPLEX, GUROBI, etc.) may allow the resolution of such optimization problems in reasonable computation times, i.e., adequate for real-time control. Typically, the train rescheduling problem statement is based on Integer Programming (IP), Mixed Integer Linear Programming (MILP), Linear Programming (LP), or Nonlinear Programming (NLP) [12]. This paper considers the use of the MILP modeling, which has been extensively tested for solving (re)scheduling problems and for which a large variety of effective solvers have been developed. Depending on the scale and level of detail to be considered in the train rescheduling problem, different types of models can be developed that are generally classified in three main classes: (1) *Macroscopic models*, i.e., a high-level representation in which stations are nodes and the connecting tracks are links between the nodes. The results of such a modeling technique can mainly be departures and arrival times, and possible routes. Then, further refinements are required before the actual application of the dispatching results to the railway system. (2) *Microscopic models*, i.e., a low-level representation of the railway system, which includes at least block sections and switch locations in the network as well as rolling stock and infrastructure availability constraints. Due to the huge amount of information included in the model, computational complexity quickly increases and can become a severe issue especially when controlling large-scale systems. (3) *Mesoscopic models*, i.e., a middle-level representation of the system, which includes elements from both the macroscopic and microscopic modeling. Some parts of the network, e.g., stations, are detailed, whereas others, such as links between stations, are modeled macroscopically.

Regarding the rescheduling of the railway traffic in case of *disruptions*, mainly macroscopic and microscopic MILP models have been used in the literature, see e.g. [13], [14]. This paper aims at filling the gap by using a mesoscopic model that, on one hand allows the reduction of the computational complexity while increasing the dimension of the problem and, on the other hand, reduces the number of eventual re-adaptations of the rescheduled timetable.

III. MILP models for train rescheduling

The rescheduling of the railway traffic can be represented as a constrained optimization problem, and the railway network as an event-driven system, whose evolution is determined by the occurrence of train runs. In this paper, both a macroscopic and a mesoscopic MILP model based on the work in [15] are considered to represent the railway network in case of full blockade between two consecutive stations. Both models can be used to represent the nominal and the disrupted functioning of the network and include recovery actions to reschedule the disrupted traffic. The macroscopic model allows quick cancellations and reorderings of train runs, short-turnings, and shunting actions in stations; the mesoscopic model allows in addition the representation of capacity limits (i.e., number of available platforms and respective assignment of trains) and ordering of trains on platforms for the disrupted stations. In case of a full blockade between two consecutive stations, the trains of the interrupted lines can enter the two stations only in one direction, then can be short-turned and used to perform outgoing train runs on the opposite direction that otherwise have to be canceled. Alternatively, shunting actions consist in moving trains to/from shunting yards when they cannot be used for short-turning. Note that, in general, shunting is avoided due to its costs in terms of time and resources.

The models are based on the concept of *train run*, which is represented as a *departure-arrival pair* $(d_i, a_i$ with $i \in T$, where T is the set of the indices of all train runs. The set T is divided into two subsets: T_{ND} consisting of all indices of train runs that are not influenced by the disruption, and set T_{D} consisting of all indices of train runs that are directly affected by the disruption and can be canceled, short-turned or shunted. A track of the network is represented by e, with $e \in E$, where E is the set containing the indices of all tracks. Each train run is associated with a track, the set of the indices of all train runs not affected by the disruption and associated with the track e is represented by $T_e \subset T_{\mathrm{ND}}$, while the set of all train runs affected by the disruption and associated with the track e is represented by $\overline{T}_e \subset T_{\text{D}}$. The dynamics of the system is described by the following constraints set:

- *Timetable constraints*

$$\begin{aligned} d_i &\geq r_{\text{d},i} \quad \forall i \in T \\ a_i &\geq r_{\text{a},i} \quad \forall i \in T \end{aligned} \tag{1}$$

  where $r_{\text{d},i}$ and $r_{\text{a},i}$ are the *nominal departure and arrival times* of the i-th train run, while d_i and a_i are the rescheduled departure and arrival times.

- *Running time constraints*

$$\begin{aligned} a_i &\geq d_i + \tau_{\text{rt},i} && \forall i \in T_{\text{ND}} \\ a_i &\geq d_i + \tau_{\text{rt},i} + \beta c_i && \forall i \in T_{\text{D}} \end{aligned} \tag{2}$$

  where $\tau_{\text{rt},i}$ is the *minimal running time duration* of the i-th train run. If the train run belongs to T_{ND}, then the arrival time has to be larger than or equal to the sum of the corresponding departure time and the minimal running time duration. Otherwise, if the train run belongs to T_{D}, then a binary *cancellation variable* c_i is associated with the train run, along with a large negative constant β. If the variable c_i assumes the value 1, there is no longer any coupling between a_i and d_i, and the train run is canceled. Note that the difference between the nominal departure and arrival times of a train run is often larger than the pure *running time* $\tau_{\text{rt},i}$ to absorb small delays:

$$r_{\text{a},i} - r_{\text{d},i} \geq \tau_{\text{rt},i} \tag{3}$$

- *Continuity constraints*

$$\begin{aligned} d_j &\geq a_i + \tau_{\text{dw},(i,j)} && \forall i,j \in T_{\text{ND}} \\ d_j &\geq a_i + \tau_{\text{dw},(i,j)} + \beta c_i && \forall j \in T_{\text{ND}}, \forall\, i \in T_{\text{D}} \\ d_j &\geq a_i + \tau_{\text{dw},(i,j)} + \beta c_j && \forall i \in T_{\text{ND}}, \forall j \in T_{\text{D}} \end{aligned} \tag{4}$$

  where $\tau_{\text{dw},(i,j)}$ is the train *dwell time* in the station connecting train run i to train run j. When both train runs belong to set T_{ND}, the departure of the j-th train run has to be larger than or equal to the sum of the arrival of the i-th train run and the dwell time. Otherwise, if the runs belong to T_{D} a cancellation variable is included in the constraints along with a large negative constant β. When the cancellation variable has the value 1, there is no longer coupling between a_i and d_j. Note that the dwell times in the timetable are the absolute minimum waiting times at stations and the difference:

$$r_{\text{d},j} - r_{\text{a},i} \geq \tau_{\text{dw},(i,j)} \tag{5}$$

  can be larger than the dwell time providing a buffer time.

- *Headway time constraints*

$$\begin{aligned}
d_k &\geq \tau_{\text{h},(k,l)} + \beta(1 - u_{(k,l)}) + d_l && \forall k,l \in T_e \\
d_l &\geq \tau_{\text{h},(k,l)} + \beta u_{(k,l)} + d_k && \forall k,l \in T_e \\
d_k &\geq \tau_{\text{h},(k,l)} + \beta(1 - u_{(k,l)} + c_k) + d_l && \forall l \in T_e, \forall k \in \overline{T}_e \\
d_k &\geq \tau_{\text{h},(k,l)} + \beta(1 - u_{(k,l)} + c_l) + d_l && \forall l \in \overline{T}_e, \forall k \in T_e \\
d_k &\geq \tau_{\text{h},(k,l)} + \beta(1 - u_{(k,l)} + c_l + c_k) + d_l && \forall l,k \in \overline{T}_e \\
d_l &\geq \tau_{\text{h},(k,l)} + \beta(u_{(k,l)} + c_k) + d_k && \forall l \in T_e, \forall k \in \overline{T}_e \\
d_l &\geq \tau_{\text{h},(k,l)} + \beta(u_{(k,l)} + c_l) + d_k && \forall l \in \overline{T}_e, \forall k \in T_e \\
d_l &\geq \tau_{\text{h},(k,l)} + \beta(u_{(k,l)} + c_l + c_k) + d_k && \forall k,l \in \overline{T}_e \\
a_k &\geq \tau_{\text{h},(k,l)} + \beta(1 - u_{(k,l)}) + a_l && \forall k,l \in T_e \\
a_l &\geq \tau_{\text{h},(k,l)} + \beta u_{(k,l)} + a_k && \forall k,l \in T_e \\
a_k &\geq \tau_{\text{h},(k,l)} + \beta(1 - u_{(k,l)} + c_k) + a_l && \forall l \in T_e, \forall k \in \overline{T}_e \\
a_k &\geq \tau_{\text{h},(k,l)} + \beta(1 - u_{(k,l)} + c_l) + a_l && \forall l \in \overline{T}_e, \forall k \in T_e \\
a_k &\geq \tau_{\text{h},(k,l)} + \beta(1 - u_{(k,l)} + c_l + c_k) + a_l && \forall k,l \in \overline{T}_e \\
a_l &\geq \tau_{\text{h},(k,l)} + \beta(u_{(k,l)} + c_k) + a_k && \forall l \in T_e, \forall k \in \overline{T}_e \\
a_l &\geq \tau_{\text{h},(k,l)} + \beta(u_{(k,l)} + c_l) + a_k && \forall l \in \overline{T}_e, \forall k \in T_e \\
a_l &\geq \tau_{\text{h},(k,l)} + \beta(u_{(k,l)} + c_l + c_k) + a_k && \forall k,l \in \overline{T}_e
\end{aligned} \tag{6}$$

  where $\tau_{\text{h},(k,l)}$ is the *headway time* between two departures (arrivals) of train runs, *k* and *l*, which run on the same track *e*; while $u_{(k,l)}$ is the binary *headway ordering variable*. If the train runs belong to $T_e \subset T_{\text{ND}}$, no cancellation variable is present in the formulation and two alternatives are considered: (1) if $u_{(k,l)} = 1$, then the departure (arrival) of train run k has to be larger than or equal to the sum of the departure (arrival) of train run l and the headway time $\tau_{\text{h},(k,l)}$, otherwise (2) if $u_{(k,l)} = 0$, the departure of train run l has to be larger than or equal to the sum of the departure of train run k and the headway time $\tau_{\text{h},(k,l)}$. If the train runs belong to $\overline{T}_e \subset T_{\text{D}}$, a cancellation variable c_i is included in the constraints along with a large negative constant β. When the cancellation variable takes the value 1, there is no longer any coupling between d_k (a_k) and d_l (a_l).

- *Short-turn constraints*
  Consider a station $s \in S_{\text{ST}}$, where S_{ST} is the set of stations where trains can be short-turned during the full blockade, and two train runs i and j with the respective proceeding train run $q(i)$ and preceding train run $p(j)$. If $q(i)$ and $p(j)$ are canceled (i.e., $c_{q(i)} = 1$ and $c_{p(j)} = 1$), the arrival a_i with $i \in I_s$ (where I_s is the set of incoming train runs in station s during the full blockade) can be combined in station s with the departure d_j with $j \in O_s$ (where O_s is the set of outgoing train runs from s during the full blockade) so that continuity is ensured to the transportation service.
  The short-turn is then modeled with the following constraint:

$$\begin{gathered} d_j \geq a_i + \tau_{\text{turn},(i,j)} + \beta(3 - c_{p(j)} - c_{q(i)} - b_{(i,j)}) \\ \forall i \in I_s, \forall j \in O_s, \forall s \in S_{\text{ST}} \end{gathered} \tag{7}$$

$$\left.\begin{aligned} \sum_{i \in I_s} b_{(i,j)} + c_j + (1 - c_{p(j)}) = 1 \quad \forall j \in O_s \\ \sum_{j \in O_s} b_{(i,j)} + c_i + (1 - c_{q(i)}) = 1 \quad \forall i \in I_s \end{aligned}\right\} \forall s \in S_{\text{ST}} \tag{8}$$

  where $\tau_{\text{turn},(i,j)}$ is the *short-turn time*, i.e., the time necessary for the short-turn operation, and $b_{(i,j)}$ is the binary *short-turn variable*. The inequality constraint (7) imposes that train run j and i can be connected only if both are not canceled and if the short-turn variable $b_{(i,j)}$ is equal to 1, then the departure d_j can take place only after the arrival a_i has occurred and $\tau_{\text{turn},(i,j)}$ has elapsed. Furthermore, the equality constraints (8) impose that each arrival should be assigned to a unique departure and vice versa.

- *Shunting constraints*
Consider a station $s \in S_{\text{ST,S}}$ where $S_{\text{ST,S}}$ is the set of stations where trains can be short-turned or shunted from/to the shunting yard. The following *shunting variable* is introduced: $y_{\text{in},i} \in \{0,1\}$, $i \in I_s$ which is used to assign to a planned departure the rolling stock in the shunting yard, and $y_{\text{out},j} \in \{0,1\}$, $j \in O_s$ which is used for shunting the rolling stock of an arriving train. Then, it holds that

$$\left.\begin{array}{l} y_{\text{in},i} + (1 - c_{q(i)}) + c_i + \sum_{j \in O_s} b_{(i,j)} = 1 \quad \forall i \in I_s \\ y_{\text{out},j}(1 - c_{p(j)}) + c_j + \sum_{i \in I_s} b_{(i,j)} = 1 \quad \forall j \in O_s \end{array}\right\} \forall s \in S_{\text{ST,S}} \tag{9}$$

The macroscopic model is then composed by the constraint sets (1) to (9), which can characterize the nominal and the disrupted behavior of the system neglecting some practical and essential restrictions of the real system. Such a limitation is here overcome by extending the macroscopic model to a mesoscopic representation of the system. In particular, the following constraint sets are included:

- *Capacity constraints*
In the following constraints, the capacity limit of the stations at each end of the disruption is taken into account. In particular, a *short-turn on platform variable* $b_{p,(i,j)} \in \{0,1\}$ is introduced, with $i \in I_s$, $j \in O_s$, and $p \in P_s$, where P_s is the set of platforms for the considered station $s \in S_{\text{ST,S}}$. Then the following capacity constraint holds:

$$b_{(i,j)} = \sum_{p \in P_s} b_{p,(i,j)}, \text{ with } s \in S_{\text{ST,S}} \tag{10}$$

meaning that a short-turn can be assigned to only one platform.

- *Ordering constraints*
If arrival a_i with $i \in I_s$ is connected to departure d_j with $j \in O_s$ and assigned to platform $p \in P_s$ ($b_{p,(i,j)} = 1$) and arrival a_k with $k \in I_s$ is connected to departure d_l with $l \in O_s$ and assigned also to platform $p \in P_s$ ($b_{p,(k,l)} = 1$), it is necessary to decide their order. Then, the *ordering on platform variable* $\omega_{(x,y)}$ is introduced, whose value is set to one when the arrival a_y with $y \in I_s$, has to be scheduled after the departure d_x with $x \in O_s$.
Hence, the ordering constraints are modeled as follows:

$$\begin{aligned} a_i &\geq d_l + \tau_{\text{ord},(i,l)} + \beta(1 - \omega_{(l,i)}) \\ a_k &\geq d_j + \tau_{\text{ord},(k,j)} + \beta(1 - \omega_{(j,k)}) \end{aligned} \tag{11}$$

$$\begin{aligned} &-1 + \beta(2 - b_{p,(i,j)} - b_{p,(k,l)}) + \omega_{(l,i)} + \omega_{(j,k)} \geq 0 \\ &\forall i,k \in I_s, \forall j,l \in O_s, p \in P_s, \forall s \in S_{\text{ST,S}} \end{aligned} \tag{12}$$

where $\tau_{\text{ord},(i,l)}$ is the *minimum ordering time* imposed between arrival a_i and departure d_l. The same holds for$\tau_{\text{ord},(i,l)}$. The platform ordering constraint (12) imposes that if the short-turns (i,j) and (k,l) are assigned to the same platform p, then only one ordering variable among $\omega_{(l,i)}$ and $\omega_{(j,k)}$ can assume the value 1.

Then the mesoscopic model consists in the constraint sets from (1) to (12). Given the railway system, the MILP rescheduling problem (both in the macroscopic and in the mesoscopic fashion) can be written in the standard form as follows:

$$\begin{aligned} &\text{minimize} \quad f = \mathbf{g}^{\text{T}}\mathbf{x} \\ &\text{subject to} \quad \mathbf{A}\mathbf{x} \leq \mathbf{z} \end{aligned} \tag{13}$$

with $\mathbf{g}$ a constant *weight vector* and $\mathbf{x}$ the ***decision variables vector***. The elements of the weight vector $\mathbf{g}$ can assume different values depending on the purpose of the optimization.

More in detail, the MILP problem based on the *mesoscopic model* can be written as follows:

$$\begin{aligned} &\text{minimize} \quad f = \mathbf{g}^{\text{T}}\mathbf{x} \\ &\text{subject to} \quad \mathbf{A}_1\mathbf{x} \leq \mathbf{z}_1 \end{aligned} \tag{14}$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{d}^{\text{T}} & \mathbf{a}^{\text{T}} & \mathbf{c}^{\text{T}} & \mathbf{u}^{\text{T}} & \mathbf{b}^{\text{T}} & \mathbf{y}^{\text{T}} & \boldsymbol{\omega}^{\text{T}} \end{bmatrix}^{\text{T}}$ is the decision variables vector and includes the departure, arrival, cancellation, headway ordering, short-turn, short-turn on platform, and ordering on platform variables. The constraints set includes equations (1) to (12), i.e., the timetable, running time, continuity, headway time, short-turn, shunting, capacity, and ordering constraints.

The MILP problem based on the *macroscopic model* can be written as:

$$\begin{aligned} &\text{minimize} \quad f = \mathbf{g}^{\text{T}}\mathbf{x} \\ &\text{subject to} \quad \mathbf{A}_2\mathbf{x} \leq \mathbf{z}_2 \end{aligned} \tag{15}$$

where the constraints set includes equations (1) to (9), i.e., the timetable, running time, continuity, headway time, short-turn, and shunting constraints. Note that for the macroscopic optimization problem the short-turn on platform and the ordering variables are canceled.

IV. The Bi-Level Algorithm

The bi-level algorithm consists in two consecutive steps (i.e., Step 1 and Step 2) in which the two MILP problems presented in the previous section are sequentially solved and the rescheduled timetable for the mesoscopic problem is obtained. In particular, in Step 1 the rescheduling horizon is set, the macroscopic MILP problem (15) is solved and the optimal decision variables vector $\bar{\mathbf{x}}$ is obtained. Then, in Step 2 the mesoscopic MILP problem (14) is set and the cancellations and non-admissible short-turnings of Step 1 are used to simplify its resolution, by reducing the search space. In particular, the mesoscopic model is modified by adding to the constraint set a number of $n+m$ equality constraints. The first n equality constraints assign the value 1 to the cancellation variables that assume the value 1 in the vector $\bar{\mathbf{x}}$. The remaining m equality constraints assign the value 0 to the short-turn variables that assume the value 0 in the results vector $\bar{\mathbf{x}}$. Consequently also the corresponding short-turns on platform variables are set to 0. In other words, the cancellations of train runs assigned in Step 1 are kept in the optimization problem of Step 2 and the short-turns of train runs that are not-admissible in Step 1 are kept infeasible in the optimization problem of Step 2. The optimization stops when a stop criterion is satisfied, e.g., the difference between two consecutive solutions is lower than a certain value Δ.

V. A CASE STUDY

In this section the bi-level algorithm is used for the rescheduling of the railway traffic in the Dutch railway network. The occurrence of a full blockade between the two consecutive stations Dordrecht (Dor) and Lage-Zwaluwe (Lzw) is here considered, where the Moerdijk Bridge is often blocked due to recurrent adverse weather conditions. Figure 1 shows the lines of the network (black lines – tot. 76), the stations (blue circles – tot. 66), the area that is directly affected by the disruption and where trains can be canceled, short-turned, or shunted (white lines), and the disrupted train section (signaled with a red cross). In the case of a full blockade, trains can be short-turned in Dordrecht (Lage-Zwaluwe) and passengers travel by bus from this station to Lage-Zwaluwe (Dordrecht) during the disruption. The nominal timetable considered for the case-study regards all train lines that run during the afternoon of a weekday, excluding extra runs for rush-hours. Table I reports the train lines considered in the case study that are directly affected by the blockade: all of them run twice every hour in both directions. Due to capacity limitations at the turning stations Dordrecht (six platforms) and Lage-Zwaluwe (four platforms) or because some trains cannot be turned for a return trip, trains of affected lines might need to be canceled before reaching their final destination. In the case study the trains of the affected lines (white lines in Fig. 1) can be canceled. Further away from the disruption, trains from these lines must keep running just as trains from all other lines.

The bi-level algorithm is then executed considering the following rescheduling problem settings: simulation window SW=300 min; rescheduling horizon RH=200 min; disruption length $t_{\mathrm{disr}} = 120$ min (disruption start time $t_{\mathrm{ds}} = 100$ and disruption end time $t_{\mathrm{de}} = 220$); recovery time after the disruption of 30 minutes; headway time $\tau_{\mathrm{h}} = 3$ min; minimum time for a short turning $\tau_{\mathrm{turn}} = 5$ min; ordering time $\tau_{\mathrm{ord}} = 3$ min; minimum dwell time $\tau_{\mathrm{dw}} = 2$ min; constant $\beta = -1000$, stop criterion $\Delta = 0.01$. Here, the focus is on the minimization of the delays spreading over the network, as well as the minimization of the cancellations and of the shunting actions. Hence, the constant vector is set as $\mathbf{g} = \begin{bmatrix} \mathbf{1}^{\mathrm{T}} & \mathbf{1}^{\mathrm{T}} & \boldsymbol{\lambda}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & \mathbf{0} & \boldsymbol{\gamma}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ where the weight vector $\lambda = 100$ for the canceling variables and $\gamma = 250$ for the shunting variables, so as to minimize cancellations and shunting actions. Note that each element of the vector $\mathbf{g}$ is still a vector and has the same dimension as the corresponding decision variable vector in $\mathbf{x}$.

The dimensions, in terms of number of constraints and variables, of the two MILP problems are reported in Table II. The results of the algorithm are presented in Figures 2 to 4. Figure 2 reports the rescheduled timetable for a significant portion of the disrupted zone. In particular, it shows a time-distance diagram of all train runs on the route from stations Den Haag CS to Venlo, i.e., a route that is shared among all of the disrupted lines listed in Table I (IC1900 in orange, IC2151 in dashed orange, IC2249 in green, SPR 5000 in light blue, SPR 5100 in blue) and lines IC3600 (Roosendaal-Zwolle in light violet), IC5200 (Tilburg-Eindhoven in violet), IC600 (Amsterdam CS-Rotterdam CS, in light green), IC2138 (Amsterdam CS-Dordrecht in red), and SPR6000 (Breda-Utrecht, in teal). Trains that overtake each other are allowed because of multiple tracks available between stations. Train runs that take place between minute 0 and 100 are shown in grey and are not involved in the rescheduling. Short-turnings and assignments between arrival and departure events are shown in brown. Figure 3 shows the schedule for the six platforms of station Dordrecht while Figure 4 shows the schedule for the four platforms of station Lage-Zwaluwe. It is worth noting that for both stations a feasible schedule is found with no overlaps between consecutive dwell periods and respecting the headway time dwells. Moreover, it has to be noticed that the rescheduling process is not limited to the disruption period but continues until the

TABLE I
LINES AFFECTED BY THE DISRUPTION.

Line	Origin	Destination	Times/hr	Blocked
IC1900	Den Haag	Venlo	2	Yes
IC2151	Amsterdam CS	Vlissingen	2	Yes
IC2249	Amsterdam CS	Dordrecht	2	No
SPR5000	Den Haag CS	Breda	2	Yes
SPR5100	Den Haag CS	Roosendaal	2	Yes

Fig. 1. Main Dutch railway network and disruption (red cross). DH CS=Den Haag CS, Rot=Rotterdam, Dor=Dordrecht, Lzw= Lage Zwaluwe, Rsd=Rosendaal, Bre=Breda, Ehv=Eindhoven, Vnl=Venlo.

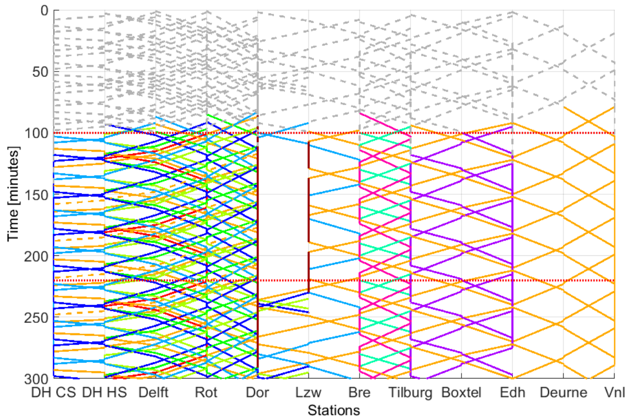

Fig. 2. The rescheduled graphical timetable.

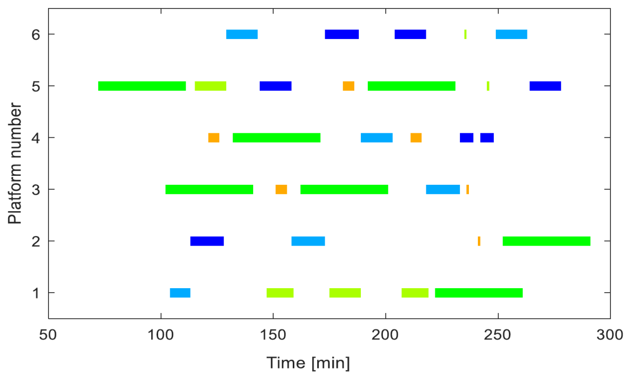

Fig. 3. The platform assignment in station Dordrecht.

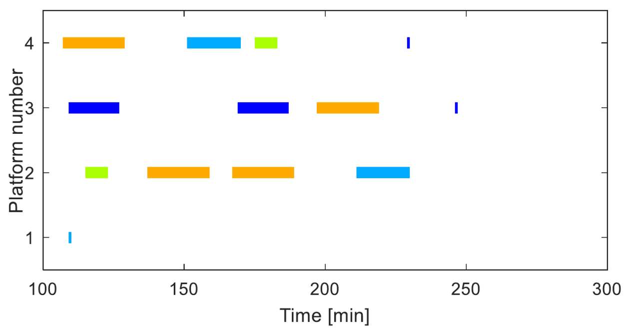

Fig. 4. The platform assignment in station Lage Zwaluwe.

nominal timetable is again suitable, thanks to the inclusion of the recovery time.

The problem is solved with the Gurobi solver in 90 s, on an Intel Quadcore 2.4 Ghz and 8 Gb Ram. The resulting computation time can be further reduced by considering an optimization of the code and/or its execution on a dedicated and more performant computer. Nevertheless, the computation time is significantly less than the 18 minutes time span necessary for the resolution of the full mesoscopic MILP problem without the bi-level algorithm. The resulting arrival delay is on average equal to 4.71 minutes while the maximum arrival delay is equal to 7.00 minutes. The percentage of delayed train runs stands at 2.62% over 1070 total train runs, while the number of canceled train runs is equal to 0 and the number of shunting actions is equal to 1. It has also to be noticed that the resolution with the bi-level algorithm leads to results comparable to that of the full mesoscopic problem both in terms of delays, percentage of delayed train runs, and cancellations. This reveals that, despite the capacity constraints, the method results in a high performance of the system.

TABLE II

MILP PROBLEMS DIMENSIONS.

Macroscopic MILP model	
No. of constraints	90124
No. of variables	9719
Mesoscopic MILP model	
No. of constraints	187344
No. of variables	11201

VI. CONCLUSIONS

In this paper it is proposed a bi-level railway rescheduling algorithm that allows obtaining a feasible rescheduled timetable in a short computation time in case of disruption based on mesoscopic modeling. The method has been tested on a disruption scenario for the Dutch railway network and results are promising both in terms of delay minimization and computation time. Further research will explore the robustness of the technique to various types of disruptions. It shall also extend the method to the case of multiple simultaneous disruptions and/or to the case of multiple consecutive stations involved in a single disruption.

REFERENCES

[1] M. Dotoli, N. Epicoco, M. Falagario, B. Turchiano, G. Cavone, and A. Convertini, “A Decision Support System for real-time rescheduling of railways,” in 2014 European Control Conference, ECC 2014, 2014, pp. 696–701.

[2] G. Cavone, M. Dotoli, N. Epicoco, and C. Seatzu, “Intermodal terminal planning by Petri Nets and Data Envelopment Analysis,” Control Eng. Pract., vol. 69, 2017.

[3] B. Kersbergen, T. van den Boom, and B. De Schutter, “Distributed model predictive control for railway traffic management,” Transp. Res. Part C Emerg. Technol., vol. 68, pp. 462–489, Jul. 2016.

[4] X. Li, B. Shou, and D. Ralescu, “Train rescheduling with stochastic recovery time: A new track-backup approach,” IEEE Trans. Syst. Man, Cybern. Syst., vol. 44, no. 9, pp. 1216–1233, Sep. 2014.

[5] V. Cacchiani et al., “An overview of recovery models and algorithms for real-time railway rescheduling,” Transp. Res. Part B: Methodological, vol. 63. Pergamon, pp. 15–37, May-2014.

[6] T. Dollevoet, D. Huisman, L. G. Kroon, L. P. Veelenturf, and J. C. Wagenaar, “Application of an iterative framework for real-time railway rescheduling,” Comput. Oper. Res., vol. 78, pp. 203–217, Feb. 2017.

[7] W. Fang, S. Yang, and X. Yao, “A Survey on Problem Models and Solution Approaches to Rescheduling in Railway Networks,” IEEE Trans. on Intell. Transp. Sys., vol. 16, no. 6. pp. 2997–3016, Dec-2015.

[8] J. Jacobs, "Reducing delays by means of computer-aided 'on-the-spot' rescheduling," Adv. Transp., vol. 15, pp. 603–612, May 2004.

[9] P. Pellegrini, G. Marliere, R. Pesenti, and J. Rodriguez, "RECIFE-MILP: An Effective MILP-Based Heuristic for the Real-Time Railway Traffic Management Problem," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 5, pp. 2609–2619, Oct. 2015.

[10] P. Pellegrini, G. Marlière, and J. Rodriguez, "Real time railway traffic management modeling track-circuits," OpenAccess Ser. Informatics, vol. 25, pp. 23–34, Jan. 2012.

[11] J. Törnquist and J. A. Persson, "N-tracked railway traffic re-scheduling during disturbances," Transp. Res. Part B Methodol., vol. 41, no. 3, pp. 342–362, Mar. 2007.

[12] J. Törnquist, "Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms," Algorithmic MeThods Model. Optim. Railways, vol. 2, p. 23p, 2006.

[13] S. Narayanaswami and N. Rangaraj, "Modelling disruptions and resolving conflicts optimally in a railway schedule," Comput. Ind. Eng., vol. 64, no. 1, pp. 469–481, Jan. 2013.

[14] N. Ghaemi, O. Cats, and R. M. P. Goverde, "A microscopic model for optimal train short-turnings during complete blockages," Transp. Res. Part B Methodol., vol. 105, pp. 423–437, Nov. 2017.

[15] L. L. Blenkers, T. J. J. van den Boom, and B. Kersbergen, "An exploratory study on railway disruption management using switching max-plus linear models," in Rail Lille - 7th International Conference on Railway Operations Modelling and Analysis, 2017, pp. 334–352.