Delft University of Technology

Delft Center for Systems and Control

Technical report 21-002

An MPC-based rescheduling algorithm for disruptions and disturbances in large-scale railway networks*

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

If you want to cite this report, please use the following reference instead:
G. Cavone, T. van den Boom, L. Blenkers, M. Dotoli, C. Seatzu, and B. De Schutter, "An MPC-based rescheduling algorithm for disruptions and disturbances in large-scale railway networks," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, p. 99-112, Jan. 2022. doi:10.1109/TASE.2020.3040940

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

An MPC-based rescheduling algorithm for disruptions and disturbances in large-scale railway networks

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

***Abstract*—Railways are a well-recognized sustainable transportation mode that helps to satisfy the continuously growing mobility demand. However, the management of railway traffic in large-scale networks is a challenging task, especially when both a major disruption and various disturbances occur simultaneously. We propose an automatic rescheduling algorithm for real-time control of railway traffic that aims at minimizing the delays induced by the disruption and disturbances, as well as the resulting cancellations of train runs and turn-backs (or short-turns) and shuntings of trains in stations. The real-time control is based on the Model Predictive Control (MPC) scheme where the rescheduling problem is solved by Mixed Integer Linear Programming using macroscopic and mesoscopic models. The proposed resolution algorithm combines a distributed optimization method and a bi-level heuristics to provide feasible control actions for the whole network in short computation time, without neglecting physical limitations nor operations at disrupted stations. A realistic simulation test is performed on the complete Dutch railway network. The results highlight the effectiveness of the method in properly minimizing the delays and rapidly providing feasible feedback control actions for the whole network.**

***Index Terms*—Rescheduling algorithms, Model Predictive Control, Mixed Integer Linear Programming, Railway traffic disruption.**

***Note to Practitioners*—This paper aims at contributing to the enhancement of the core functionalities of Automatic Train Control (ATC) systems and, in particular, of the Automatic Train Supervision (ATS) module, which is included in ATC systems. In general, the ATS module allows to automate the train traffic supervision and consequently the rescheduling of the railway traffic in case of unexpected events. However, the implementation of an efficient rescheduling technique that automatically and rapidly provides the control actions necessary to restore the railway traffic operations to the nominal schedule is still an open issue. Most literature contributions fail in providing rescheduling methods that successfully determine high-quality solutions in less than one minute and include real-time information regarding the large-scale railway system state. This research proposes a semi-heuristic control algorithm based on MPC that, on the one hand, overcomes the limitations of manual rescheduling (i.e., suboptimal, stressful, and delayed decisions), and on the other hand, offers the advantages of on-line and closed-loop control of railway traffic based on continuous monitoring of the traffic state to rapidly restore railway traffic operations to the nominal schedule. The semi-heuristic procedure permits to significantly reduce the computation time necessary to solve the rescheduling problem compared to an exact procedure; moreover, the use of a distributed optimization approach permits the application of the algorithm to large instances of the rescheduling problem, and the inclusion of both the traffic and rolling stock constraints related to the disrupted area. The method is tested on a realistic simulation environment, thus still requires further refinements for the integration into a real ATS system. Further developments will also consider the occurrence of various simultaneous disruptions in the network.**

This research is partially funded by the National Natural Science Foundation of China under Grant No. 61950410604.

G. Cavone and M. Dotoli are with the Department of Electrical and Information Engineering, Polytechnic of Bari, Bari, Italy (corresponding author email: 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, Delft University of Technology, 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).

I. Introduction

Railways are a well-recognized sustainable transportation modality and are among the most carbon-efficient modes of mass transportation, exhibiting the highest share of electrification [1]. Consequently, it is fundamental for railway companies to offer high-quality service standards that allow driving towards railways the modal shift of customers that care for sustainability.

This article focuses on railway traffic rescheduling in large-scale systems when both a disruption, i.e., a long interruption in the railway service, and short delays suddenly occur. In the related literature, the problem of railway traffic rescheduling has been extensively discussed [2] [3] [4] [5] [6], leading to the development of decision support systems that can model specific situations, compute optimal solutions in real time, and suggest proper control actions. However, most of the available contributions focus on the rescheduling in case of short delays (also called disturbances) [4], [5], while few works consider the problem of long unexpected disruptions of the service, which are less frequent but significantly decrease the system performance [7]. Furthermore, most contributions consider the application of only one type of control action (e.g., canceling of train runs) or simple combinations of a few of the available ones [8] [9] [10]; finally they typically use either macroscopic [9], [11], [12] or microscopic models [13], [14], where macroscopic models allow the representation of large areas neglecting important physical limitations of the network so as to reduce the problem complexity, while microscopic models keep into account a lot of details, resulting in a huge number of variables and constraints.

Here, we focus on real-time management of railway traffic in large-scale networks in case of both a disruption and small disturbances. In particular, we consider the problem of automatically rescheduling the railway traffic when a full blockade occurs between two consecutive stations, thus preventing the circulation of trains in both directions, while small disturbances are still present and interfere with the nominal scheduling for the whole network. Our approach differs from the state of the art on railway traffic management that mainly provides open-loop control techniques for the railway traffic management and looks at the global optimality instead of global feasibility of the solutions [15]. Indeed, our purpose is to reschedule on-line and in closed loop the railway traffic in a large-scale network without neglecting the capacity limitations and the actual operations to be performed in the stations at the ends of the disrupted railway section. To this aim, we present an automatic rescheduling algorithm that combines Model Predictive Control (MPC) with an integrated methodology based on a distributed optimization algorithm and a bi-level heuristics. The integrated methodology performed in the control scheme is based both on a macroscopic and a mesoscopic Mixed Integer Linear (MIL) model that properly represent the network and its behavior in nominal, disturbed, and disrupted modes. In the on-line feedback control scheme, the algorithm iteratively solves a distributed optimization problem based on the two MIL models together with a linear cost criterion, i.e., a MIL Programming (MILP) problem, and provides in real time updated control actions to be implemented in the network. The setting of the distributed optimization problem varies with each iteration of the algorithm, thus depending on the state of the network (i.e., nominal, disturbed, or disrupted) and on the state of the zones of the network (i.e., disrupted or non-disrupted). The bi-level heuristics acts only when it is necessary to reschedule traffic in the disrupted zone. In particular, the algorithm first solves the *macroscopic* MILP sub-problem related to the disrupted zone and then, based on the obtained results, solves the corresponding *mesoscopic* MILP sub-problem. This approach allows limiting the computational burden of the control scheme without neglecting physical limitations and operations to be performed at disrupted stations.

Summing up, the main contributions of this paper are: (1) the development of a bi-level heuristics for fast solution of a mesoscopic MILP rescheduling problem in case of full blockade; (2) the integration of the bi-level heuristics in a distributed optimization algorithm for fast solution of a large-scale rescheduling problem; (3) the implementation of an automatic technique that allows on-line feedback control of a large-scale railway system when both a disruption and various unexpected disturbances occur, thanks to the integration of the distributed optimization algorithm and the bi-level heuristics in an MPC control scheme.

The effectiveness of the proposed technique is tested in a simulation environment on a complex real case-study. More in detail, we apply the method to the national Dutch railway network, where we use the real timetable consisting of all train lines that run in the whole country during the afternoon of a weekday. We simulate the presence of a full blockade between two consecutive stations and the presence of randomly generated small disturbances in the rest of the network in 100 different instances. The obtained results show that the proposed method provides feasible control actions in less than 1 minute, i.e., respecting the sampling time limit of the MPC procedure, and being fully in accordance with the real-time train dispatching rules. Furthermore, performance indices highlight that the method ensures a low percentage of delayed train runs as well as average and maximum arrival delays that are significantly lower than the standard limits of the railway companies.

The paper remainder is structured as follows. Section II recalls the related state of the art and positions the paper within it. Section III presents the MILP formulation of the rescheduling problem. Section IV describes the distributed optimization algorithm for solving the large-scale MILP problem including the bi-level heuristics. Section V describes the automatic feedback control algorithm and Section VI reports the results obtained by applying the method to the case study. Finally, Section VII summarizes the contribution of the work and the possible further developments.

II. State of the Art

A. Real-time management of railway traffic

Real-time railway traffic management is a well-studied problem in the context of railway transportation, due to the substantial losses and/or gains that it can generate to companies and customers [4]. In general, the unexpected events affecting the nominal traffic can be roughly divided into two main categories: *disturbances* and *disruptions* [4]. *Disturbance*s are defined as small perturbations that can cause short delays and consequently slightly defer the nominal arrivals and departures of multiple trains in the network. Such type of events can be caused, e.g., by a minor malfunctioning in the network. Their effects can be limited by small control actions, e.g., re-timing or reordering of trains in a station. On the other hand, *disruptions* are defined as long delays that can strongly affect the nominal circulation of trains in the network. For example, an interruption of the traffic at a certain track section can reduce the capacity of the network and necessitate drastic control actions, e.g., the cancellation of train runs. Two main categories of disruptions can be identified: *full blockades* and *partial blockades*. A full blockade consists in the complete interruption of the traffic between the stations at each end of the disruption, while a partial blockade consists in a reduction of the capacity of the tracks available for the circulation between the stations at each end of the disruption (hence, the number of the blocked tracks is smaller than the total number of the available tracks at the considered section).

A further classification regards the main control actions that can be performed in case of disruptions: (1) *short-turning*, (2) *shunting* in stations, and (3) *cancellation* of train runs. More in detail, *short-turning* consists in a U-turn of the trains that enter the disrupted stations and then in the re-routing of the trains on their way back; while for *shunting*, trains are moved from the stop platform in station to a dedicated yard in order to free the space in the station and to be used for later train runs when no other train is available. *Cancellation* obviously refers to the cancellation of some trains. Note that more formal definitions of short-turn and shunting are provided in the following Section III.

Currently, in the Netherlands and in other countries like Germany, Switzerland, Denmark, and Japan, traffic controllers deal with disruptions by using predefined solutions, i.e., *contingency plans*, that are tailored to a precise disruption scenario in a precise location and that are designed manually by experienced human traffic controllers [8]. However, contingency plans cannot cover all the disruption cases and are suitable only for small portions of larger and more complex railway systems. In case no suitable contingency plan is available, human traffic controllers are required to promptly reach an agreement about a suitable plan and consequently are charged with a stressful and challenging workload. It appears then evident that there is a need for a method that supports human train controllers in rescheduling the railway traffic in real time, independently of the type of disruption, and that also provides feasible solutions for large-scale railway systems.

The analysis of the related literature allows pointing out that in general the rescheduling problem is represented as an optimization problem and the main limitation in tackling real-time rescheduling in case of disruptions and disturbances is the high level of complexity of the problem. The long computation time necessary for the solution of the problem makes the optimization-based method impractical for a real-time control environment. To avoid such an issue, on the one hand, some contributions (i.e., [9], [11], [16]–[22]) consider simplified models of the railway network, i.e., macroscopic models, so as to reduce the complexity of the problem and so as to properly represent larger areas of the network. However, such a representation increases the workload of train controllers, who have to readapt the rescheduled timetable before applying it to the real environment. On the other hand, other papers (e.g., [8], [23], [24]) propose detailed models, i.e., microscopic models that allow a realistic representation of the problem and the immediate applicability of the rescheduled timetable. However, in a real-time context, microscopic models limit the management to only small portions of the network. Hence, there is an urgent need to extend the research so as to fill the gap and to obtain a suitable technique for the rescheduling of railway traffic (1) in case of disruptions and disturbances, (2) based on realistic models, (3) in real time, and (4) for large-scale networks.

B. MPC for real-time rescheduling in large-scale networks

The various automatic real-time rescheduling approaches presented in the literature can be distinguished into static (or open-loop rescheduling) if the rescheduling is performed only once, with full information on the state of the system, and dynamic (or closed-loop rescheduling) if the rescheduling is iteratively performed and the information to use changes over time [10]. The second class of real-time control approaches are based on a general scheme that can be resumed as follows: selected data are gathered from the real world and passed to controllers that, based on certain models and rules, come up with proper control actions. The control actions are the input to steer the system to a certain desired state or performance. In general, such systems are included in iterative frameworks that adjust the forecast and the solution along time, in a closed-loop control setup inspired by rolling horizon optimization or model predictive control (MPC). Among the available dynamic real time approaches, here we focus on the MPC [25] scheme, which is effectively used to determine the optimal dispatching actions based on a prediction of the evolution of the system under control, using a model of that system [26].

As shown in [27], for large-scale railway systems a centralized MPC scheme to control the entire system will exceed the allowable calculation time and is therefore infeasible. Especially for large networks, it is instead better to split the system into several smaller sub-areas and to control them individually. However, such a distributed approach can efficiently solve the real-time rescheduling problem for large-scale railway systems only if it relies on macroscopic models of the railway system and operations. In case of microscopic models, the computational burden may become so significant that the real-time requirements cannot be fulfilled. It is also worth noting that the MPC scheme, both in its centralized and distributed version, has been mainly used to solve the rescheduling problem only in case of disturbances (see e.g., [27]–[31]).

Considering the above state of the art, it is evidently important to develop a novel approach that allows the use of the MPC scheme in a distributed fashion in the context of disturbance and disruption management in large-scale railway networks.

This article presents a novel rescheduling algorithm that allows traffic controllers to effectively cope with disturbances and disruptions in large-scale networks. The method successfully combines the bi-level resolution of a macroscopic and a mesoscopic MILP problem, representing the behavior of a railway network affected by disturbances and disruptions, with the distributed optimization of the MPC scheme, so as to overcome the reported computational issues and to provide in real time a realistic and suitable rescheduled timetable to be used for the automatic train supervision.

III. Macroscopic and Mesoscopic Models

In this section, before introducing the MILP rescheduling problem, we provide the definition of *short-turn* and *shunting* of rolling stock, which are common control actions in railway traffic management.

Given the generic i-th *train run* as a pair (d_i, a_i) of departure-arrival times between two consecutive stations (i.e., without any intermediate stop), *short-turn* and *shunting* are defined as follows:

1) *Short-turn* consists in a turn-back of the trains that are directed to the blocked tracks. The short-turn is performed at the stations located at the ends of the full blockade. In this way trains are used to perform runs in the opposite direction of their nominal one. An example of short-turn is reported in Fig. 1 and Fig. 2. More in detail, Fig. 1 represents the normal behavior of two trains arriving from opposite directions in station s. Consider two train runs i and j with the respective subsequent train run $\xi^{-1}(i)$ and preceding train run $\xi(j)$. The train run i is combined with the subsequent train run $\xi^{-1}(i)$ and the train run $\xi(j)$ is combined with the train run

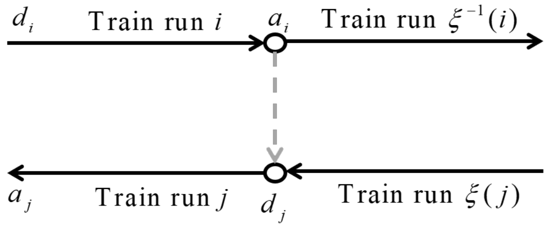

Fig. 1. Nominal traffic.

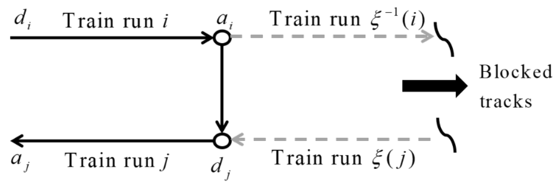

Fig. 2. Short-turn.

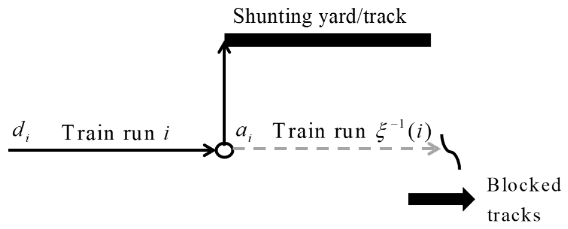

Fig. 3. Shunting a train from platform to shunting yard.

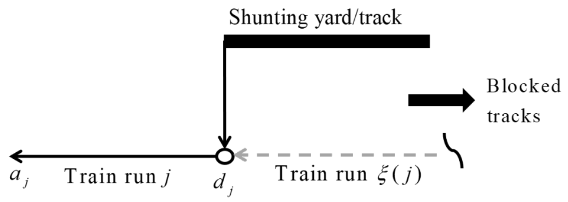

Fig. 4. Shunting a train from shunting yard to platform.

j as nominally scheduled. In practice, the two trains transit simultaneously or with a few minutes of delay, in opposite directions in the same station s. Conversely, in Fig. 2 the train run i is combined with train run j since train run $\xi^{-1}(i)$ and train run $\xi(j)$ have been canceled. In practice, the train arriving from the left to station s cannot proceed to the next station and then, after the short-turn, it performs the train run j as nominally scheduled in the opposite direction.

2) *Shunting* consists in a shift of the trains from a platform in station to a separate shunting yard/track of the same station or vice-versa. The shunting can be performed in the stations at the two ends of the full blockade to: (1) avoid their congestion/block by releasing platforms, (2) use the trains in the shunting yard/track to perform train runs that would otherwise be canceled. Examples of the two cases are reported in Figs 3 and 4. Figure 3 represents the case of a train that performs the run i but cannot proceed in its nominal direction, since the subsequent run $\xi^{-1}(i)$ has been canceled. Then the train is moved from the platform to the shunting yard in station s. Conversely, in Fig. 4 the run $\xi^{-1}(i)$ has been canceled and the run j cannot be performed by the nominal train. Then, a train is moved from the shunting yard to a platform of the station s to perform the run j.

Hereafter, we describe the formulations of the rescheduling optimization problems which are based on a macroscopic and a mesoscopic MIL model. Both optimization problems can be used in case of full blockade, delays, and simultaneous full blockade and delays. The macroscopic model mainly represents the railway network traffic, while the mesoscopic model includes also the rolling stock dispatching operations in the stations directly involved in the full blockade. For the sake of brevity, we provide only an intuitive insight of the mathematical formulations. For the formal definitions of the macroscopic and mesoscopic MIL models we refer the interested reader respectively to the works of Kersbergen *et al.* [27] and Blenkers [32] that, differently from this work, respectively provide a distributed MPC rescheduling scheme based on a macroscopic MIL model in case of delays, and a mesoscopic MIL model of the network in case of disruptions.

Let us denote the discrete time instant as$t(k)$, with $k = 0, 1, \ldots$, the constant sampling time as T_{step}, such that $t(k) = kT_{\text{step}}$, the time horizon as $T_{\text{hor}} = t(K) = KT_{\text{step}}$, the continuous (binary) decision variables vector at time $t(k)$ as $\mathbf{x}(k)$ ($\mathbf{v}(k)$). The continuous decision variable vector $\mathbf{x}(k)$ contains all time variables (departure time and arrival time) for which the times are scheduled or expected to occur in the time horizon T_{hor} at time instant $t(k)$. The binary decision variable vector $\mathbf{v}(k)$ contains all the binary variables (i.e., cancellations, headway, short-turn, shunting, and ordering variables) at time instant $t(k)$ over the time horizon T_{hor}.

The rescheduling problem can be formulated as an MIL model together with a linear cost criterion, thus leading to a MILP problem in the standard form as follows:

$$\begin{aligned} &\min_{\mathbf{x}(k),\mathbf{v}(k)} \ \mathbf{g}^{\mathrm{T}}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}(k)}\mathbf{v}(k) \\ &\text{s.t. } \mathbf{A}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{A}_{\mathbf{v}(k)}\mathbf{v}(k) \leq \mathbf{b}(k) \end{aligned} \tag{1}$$

The linear objective function $f(\mathbf{x}(k), \mathbf{v}(k)) = \mathbf{g}^{\mathrm{T}}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}(k)}\mathbf{v}(k)$ is the sum of two terms:

- the sum of the continuous decision variables vector $\mathbf{x}(k)$ weighted by the vector $\mathbf{g}^{\mathrm{T}}_{\mathbf{x}(k)}$;
- the sum of the binary decision variables vector $\mathbf{v}(k)$ weighted by the vector $\mathbf{g}^{\mathrm{T}}_{\mathbf{v}(k)}$.

The constraints set is presented in its matrix notation where:

- $\mathbf{A}_{\mathbf{x}(k)}$ is the coefficient matrix of the continuous decision variables,
- $\mathbf{A}_{\mathbf{v}(k)}$ is the coefficient matrix of the binary decision variables,
- $\mathbf{b}(k)$ is the constraint vector with known terms.

In the macroscopic and in the mesoscopic MILP problems both the objective function and the constraints sets assume different formulations as follows.

TABLE I
NOTATION

Symbol	Meaning
	INTEGER VARIABLES
$t(k)$	Time value at *k-th* sampling time
p	Index of the platform in station
$\xi(j)$	Index of the train run preceding the j-th train run
	CONTINUOUS VARIABLES
$\mathbf{d}(k)$	Rescheduled departure times vector at $t(k)$
$\mathbf{a}(k)$	Rescheduled arrival times vector at $t(k)$
$\mathbf{x}(k)$	Continuous variables vector at $t(k)$
	BINARY VARIABLES
$\mathbf{c}(k)$	Cancellation variables vector at $t(k)$
$\mathbf{u}(k)$	Headway variables vector at $t(k)$
$\mathbf{s}(k)$	Short-turn variables vector at $t(k)$
$\mathbf{s}_p(k)$	Short-turn variables vector on platform p at $t(k)$
$\mathbf{y}(k)$	Shunting variables vector at $t(k)$
$\omega(k)$	Ordering variables vector at $t(k)$
$\mathbf{v}(k)$	Binary variables vector for mesoscopic model at $t(k)$
$\mathbf{v}'(k)$	Binary variables vector for macroscopic model at $t(k)$
	CONSTANT VALUES
T_{step}	Sampling time
T_{hor}	Duration of the time window – prediction horizon
$\mathbf{g}_{\mathbf{x}(k)}$	Weight vector of the continuous variables vector $\mathbf{x}(k)$
$\mathbf{g}_{\mathbf{v}(k)}$	Weight vector of the continuous discrete vector $\mathbf{v}(k)$
$\mathbf{g}_{\mathbf{v}'(k)}$	Weight vector of the continuous discrete vector $\mathbf{v}'(k)$
$\mathbf{A}_{\mathbf{x}(k)}$	Coefficient matrix of the continuous variables vector $\mathbf{x}(k)$
$\mathbf{A}_{\mathbf{v}(k)}$	Coefficient matrix of the continuous discrete vector $\mathbf{v}(k)$
$\mathbf{A}_{\mathbf{v}'(k)}$	Coefficient matrix of the continuous discrete vector $\mathbf{v}'(k)$
$\mathbf{b}(k)$	Constraint vector with known terms at $t(k)$

In the *mesoscopic MILP problem*, the continuous decision variables vector at time instant $t(k)$ is $\mathbf{x}(k) = [\,\mathbf{d}^{\mathrm{T}}(k) \;\; \mathbf{a}^{\mathrm{T}}(k)\,]^{\mathrm{T}}$ while the binary decision variables vector is $\mathbf{v}(k) = [\,\mathbf{c}^{\mathrm{T}}(k) \;\; \mathbf{u}^{\mathrm{T}}(k) \;\; \mathbf{s}^{\mathrm{T}}(k) \;\; \mathbf{s}_p^{\mathrm{T}}(k) \;\; \mathbf{y}^{\mathrm{T}}(k) \;\; \boldsymbol{\omega}^{\mathrm{T}}(k)\,]^{\mathrm{T}}$. For the meaning of the decision variables the reader is referred to Table I.

The constraints set can be divided into the following subsets (see [27] and [32] for the mathematical description of the constraints) according to their physical meaning:

(a) *Timetable constraints,* imposing the rescheduled departure/arrival times to be equal to or higher than the nominal ones;
(b) *Running time constraints,* imposing the fulfillment of the minimum run time duration for each train run;
(c) *Dwell time constraints,* imposing the fulfillment of the minimum dwell time duration for each train stop in station;
(d) *Headway time constraints,* imposing the fulfillment of the safety time between two consecutive departures in the same direction from the same station;
(e) *Short-turn constraints,* allowing the combination of two consecutive train runs in opposite directions;
(f) *Shunting constraints,* allowing the implementation of a shunting operation (from/to platform to/from shunting yard in the disrupted stations);
(g) *Capacity constraints,* imposing the fulfillment of the capacity limits in the disrupted stations;
(h) *Ordering constraints,* imposing the fulfillment of time order constraints for the trains assigned to the same platforms in the disrupted stations.

Table II summarizes the railway traffic conditions that can be modeled with the mesoscopic MIL model, the type of control actions, and the subsets of constraints included. Hence, in the mesoscopic formulation the rescheduling problem is written as (1) subject to constraints (a)–(h) (see [32] for a detailed description of the constraints).

In the *macroscopic MILP problem*, the continuous decision variables vector at time instant $t(k)$ is still $\mathbf{x}(k) = [\,\mathbf{d}^{\mathrm{T}}(k) \;\; \mathbf{a}^{\mathrm{T}}(k)\,]^{\mathrm{T}}$ while the binary decision variable vector is $\mathbf{v}'(k) = [\,\mathbf{c}^{\mathrm{T}}(k) \;\; \mathbf{u}^{\mathrm{T}}(k) \;\; \mathbf{s}^{\mathrm{T}}(k) \;\; \mathbf{y}^{\mathrm{T}}(k)\,]^{\mathrm{T}}$, which is a sub-vector of the vector used in the mesoscopic model $\mathbf{v}(k)$, namely a vector only containing a subset of the components of $\mathbf{v}(k)$. For the meaning of the decision variables refer again to Table I. The constraints set is composed by the subsets (a) to (f) as described in Table II (see also [27]).

Table II, third and fourth rows, summarize the railway traffic conditions that can be modeled with the macroscopic MILP model, the type of control actions, and the subsets of constraints included. Note that, although the macroscopic model allows to represent the short-turn and shunting actions, it does not consider the capacity limits of the stations and the time ordering of trains on the platforms, which are instead considered in the mesoscopic one.

IV. THE DISTRIBUTED OPTIMIZATION ALGORITHM

In this section we propose the integration of a bi-level heuristics in a distributed optimization scheme with the aim of solving the macro-mesoscopic MILP rescheduling problem for the large-scale railway network over the time horizon T_{hor}. This integrated algorithm can be used only to control the system in a feedforward way, thus neglecting the real-time evolution of the system. To overcome such a limitation, in Section V we include this algorithm in an MPC scheme to obtain a feedback control approach.

Let us consider a complex large-scale rescheduling MILP problem (1) for which the dimension of the vector $[\,\mathbf{x}^{\mathrm{T}}(k) \;\; \mathbf{v}^{\mathrm{T}}\,]^{\mathrm{T}}$ is in the range 5000–15000, the dimension of the vector $\mathbf{b}(k)$ is in the range 50000–150000, and the considered time horizon $T_{\text{hor}} \leq 120$ min. We set the sampling time $T_{\text{step}} = 1$ min, which is the minimum time unit of the departure and arrival times, so as to have a sufficiently high resolution of the state of the system. We remark that the considered values of the time horizon and the time step are commonly accepted values of these parameters, see the discussion in [15]. Obviously, the proposed algorithm parameters may be straightforwardly changed if necessary.

The proposed integrated algorithm, here named *Distributed Optimization Algorithm*, is based on an iterative procedure and consists in the following two subsequent phases: configuration and core algorithm. Hereafter, the algorithm is described in detail, while Fig. 5 shows the corresponding pseudocode.

Phase 1 – CONFIGURATION

P1.1 Global Macroscopic Problem Definition

TABLE II
MAIN FEATURES OF MACROSCOPIC AND MESOSCOPIC MODELS

Type of model	Railway traffic conditions	Type of control actions	Constraints involved
Mesoscopic model	Small disturbances, full blockade, and station capacity limits	• Reordering of trains on tracks • Cancellations • Short-turn • Shunting • Platforms assignment in station • Ordering of trains in station	(a) to (h)
Macroscopic model	Small disturbances and full blockade	• Reordering of trains on tracks • Cancellations • Short-turn • Shunting	(a) to (f)
Macroscopic model	Small disturbances	• Reordering of trains on tracks	(a) to (d)

Configure the global macroscopic MILP problem:

$$\begin{aligned} \min_{\mathbf{x}(k),\mathbf{v}'(k)} \; & f(\mathbf{x}(k),\mathbf{v}'(k)) = \mathbf{g}^{\mathrm{T}}_{\mathbf{x}(k)}\mathbf{x}(k) + \; \mathbf{g}^{\mathrm{T}}_{\mathbf{v}'(k)}\mathbf{v}'(k) \\ \text{s.t. } & \mathbf{A}'_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{A}'_{\mathbf{v}'(k)}\mathbf{v}'(k) \le \mathbf{b}'(k) \end{aligned} \tag{2}$$

with

$$\begin{aligned} \mathbf{x}(k) &= [\, \mathbf{d}^{\mathrm{T}}(k) \;\; \mathbf{a}^{\mathrm{T}}(k)\,]^{\mathrm{T}} \in \mathbb{R}^{l\times 1} \\ \mathbf{v}'(k) &= [\, \mathbf{c}^{\mathrm{T}}(k) \; \mathbf{u}^{\mathrm{T}}(k) \; \mathbf{s}^{\mathrm{T}}(k) \; \mathbf{y}^{\mathrm{T}}(k)\,]^{\mathrm{T}} \in \{0,1\}^{m\times 1} \\ \mathbf{g}_{\mathbf{x}(k)} &= \mathbf{1}_{l\times 1} \\ \mathbf{g}_{\mathbf{v}'(k)} &= [\, \boldsymbol{\lambda} \;\; \mathbf{0} \;\; \mathbf{0} \;\; \boldsymbol{\gamma}\,]_{m\times 1} \end{aligned}$$

where $|\boldsymbol{\lambda}| = |\mathbf{c}(k)| = m_\lambda$ and $\boldsymbol{\lambda} = 500 \cdot \mathbf{1}_{m_\lambda\times 1}$, while $|\boldsymbol{\gamma}| = |\mathbf{y}(k)| = m_\gamma$ and $\boldsymbol{\gamma} = 1000 \cdot \mathbf{1}_{m_\gamma\times 1}$. In other words, the weights are chosen to minimize the delays in the network, as well as the cancellations and the shunting actions due to their cost in terms of time and operators. The constraints set satisfies the specifications reported in the third row of Table II.

The *parameters of the MILP problem* are set in accordance with the specifications of the real system, i.e., the *nominal timetable*, the *headway time*, the *dwell* time, the *runtimes*, the *short-turn* time, the *shunting* time, and the *ordering* time.

P1.2 Macroscopic Problem Partitioning

Partition the global macroscopic MILP problem into a number $\alpha = |Z|$ of macroscopic MILP sub-problems, with Z the set of zones:

$$\begin{aligned} \min_{\mathbf{x}(k),\mathbf{v}'(k)} \; & \mathbf{g}^{\mathrm{T}}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}'_z(k)}\mathbf{v}'_z(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}'_{z'}(k)}\mathbf{v}'_{z'}(k) \\ \text{s.t. } & \mathbf{A}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{A}_{\mathbf{v}'_z(k)}\mathbf{v}'_z(k) + \mathbf{A}_{\mathbf{v}'_{z'}(k)}\mathbf{v}'_{z'}(k) \le \mathbf{b}(k) \\ & \text{with } z \in Z \end{aligned} \tag{3}$$

Note that $\mathbf{v}'_z(k) = [\, \mathbf{c}^{\mathrm{T}}_z(k) \;\; \mathbf{u}^{\mathrm{T}}_z(k) \;\; \mathbf{s}^{\mathrm{T}}_z(k) \;\; \mathbf{y}^{\mathrm{T}}_z(k)\,]^{\mathrm{T}}$ is the vector of the binary variables of zone z, and $\nu'_{z'}(k) = [\, \mathbf{c}^{\mathrm{T}}_{z'}(k) \;\; \mathbf{u}^{\mathrm{T}}_{z'}(k) \;\; \mathbf{s}^{\mathrm{T}}_{z'}(k) \;\; \mathbf{y}^{\mathrm{T}}_{z'}(k)\,]^{\mathrm{T}}$ is the vector of the binary variables of all other zones z' different from z.

For the optimal partitioning of the global *macroscopic MILP problem*, we use the procedure by Kersbergen *et al.* [27]. This procedure consists in: i) grouping the variables and constraints of the global problem per track and choosing the number α of zones; ii) solving a Mixed Integer Quadratic Programming problem that minimizes: (1) the number of constraints that couple the sub-problems, (2) the difference in the number of constraints among the sub-problems, by keeping the disrupted area into a single zone. Note that there is no specific rule for the choice of α, but in general the higher the number of zones is, the lower the computation time needed to solve the MILP problem. For further details we refer the interested reader to [27].

P1.3 Mesoscopic Problem Definition

Set the mesoscopic MILP sub-problem for the disrupted zone z:

$$\begin{aligned} \min_{\mathbf{x}(k),\mathbf{v}(k)} \; & \mathbf{g}^{\mathrm{T}}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}_z(k)}\mathbf{v}_z(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}_{z'}(k)}\mathbf{v}_{z'}(k) \\ \text{s.t. } & \mathbf{A}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{A}_{\mathbf{v}_z(k)}\mathbf{v}_z(k) + \mathbf{A}_{\mathbf{v}_{z'}(k)}\mathbf{v}_{z'}(k) \le \mathbf{b}(k) \\ & \text{with } z \in Z \end{aligned} \tag{4}$$

with $\mathbf{v}_z(k) = [\, \mathbf{c}^{\mathrm{T}}_z(k) \;\; \mathbf{u}^{\mathrm{T}}_z(k) \;\; \mathbf{s}^{\mathrm{T}}_z(k) \;\; \mathbf{s}^{\mathrm{T}}_{p,z}(k) \;\; \mathbf{y}^{\mathrm{T}}_z(k) \;\; {}^{\mathrm{T}}_z(k)\,]^{\mathrm{T}}$ the vector of the binary variables for the disrupted zone z. The constraints set respects the specifications in Table II.

P1.4 Stop Criterion Definition

Let us define the iteration counter for the algorithm by $= 1, 2, \ldots$. The algorithm stops if $(n \ge \mathrm{N}_{\max})$ OR $(\Delta \le \Delta_{\max})$ OR $(C_{\mathrm{time}} \ge C_{\max})$, where $\Delta_{\max}$ and $C_{\max}$ provide an upper and a lower bound, respectively, to Δ and C_{time}, which are defined as follows: Δ is the error between the solution of the global problem of phase *P1.1* at the n-th generic iteration $f^n(\mathbf{x}(k), \mathbf{v}'(k))$ and the solution of the global problem of phase *P1.1* at the previous $n-1$-th iteration $f^{n-1}(\mathbf{x}(k), \mathbf{v}'(k))$; C_{time} is the total computation time until now. In order to apply the algorithm in real time, reasonable values for the stopping criterion parameters are $\mathrm{N}_{\max} = 7$, $\Delta_{\max} = 0.01$, and $C_{\mathrm{time}} = 1\,\mathrm{min}$.

Phase 2 – CORE ALGORITHM

P2.1 Set Counters - Set Initial Solution

Set the counter of the zones $z = 1$ and the iteration counter $n = 1$.

Set the initial solution

$$\bar{\mathbf{x}}(k) = [\, \mathbf{d}^{\mathrm{T}}_{\mathrm{nom}}(k) \;\; \mathbf{a}^{\mathrm{T}}_{\mathrm{nom}}(k)\,]^{\mathrm{T}} \;\text{ and }\; \bar{\mathbf{v}}'(k) = \mathbf{0}_{m\times 1}$$

Hence, we assign to the continuous decision variable the nominal departure and arrival times and to the binary variables the value 0, i.e., no control action is performed.

P2.2 Distributed Optimization

Iterate until $(n \ge \mathrm{N}_{\max})$ OR $(\Delta \le \Delta_{\max})$ OR m$(C_{\mathrm{time}} \ge C_{\max})$

Iterate until $z < \alpha$

P2.2.1 Check Zone

If z is the disrupted zone, execute the *BI-LEVEL HEURISTICS*

FIRST LEVEL

Solve the macroscopic MILP sub-problem for zone z:

$$\begin{aligned} \min_{\mathbf{x}(k),\mathbf{v}'(k)} \quad & \mathbf{g}^{\mathrm{T}}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}'_z(k)}\mathbf{v}'_z(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}'_{z'}(k)}\mathbf{v}'_{z'}(k) \\ \text{s.t.} \quad & \mathbf{A}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{A}_{\mathbf{v}'_z(k)}\mathbf{v}'_z(k) + \mathbf{A}_{\mathbf{v}'_{z'}(k)}\mathbf{v}'_{z'}(k) \leq \mathbf{b}(k) \\ & \mathbf{v}'_{z'}(k) = \bar{\mathbf{v}}'_{z'}(k) \end{aligned} \tag{5}$$

with $\bar{\mathbf{v}}'_{z'}(k)$ the solution vector of the binary variables for the zones z' obtained at the previous iteration $n-1$. At the first iteration $\bar{\mathbf{v}}'_{z'}(k)$ is the zero vector.

Let us denote the solution vector of the binary variables of (5) by $\widehat{\mathbf{v}}'(k)$. Then, we extract from $\widehat{\mathbf{v}}'(k)$ the sub-vector of the cancellation variables $\widehat{\mathbf{c}}(k)$ and the sub-vector of the short-turn variables $\widehat{\mathbf{s}}(k)$.

END FIRST LEVEL

SECOND LEVEL

Solve the mesoscopic MILP sub-problem for zone z:

$$\begin{aligned} \min_{\mathbf{x}(k),\mathbf{v}(k)} \quad & \mathbf{g}^{\mathrm{T}}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}_z(k)}\mathbf{v}_z(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}_{z'}(k)}\mathbf{v}_{z'}(k) \\ \text{s.t.} \quad & \mathbf{A}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{A}_{\mathbf{v}_z(k)}\mathbf{v}_z(k) + \mathbf{A}_{\mathbf{v}_{z'}(k)}\mathbf{v}_{z'}(k) \leq \mathbf{b}(k) \\ & \mathbf{v}'_{z'}(k) = \bar{\mathbf{v}}'_{z'}(k) \\ & \widehat{\mathbf{c}}_z(k) \circ \mathbf{c}_z(k) = \widehat{\mathbf{c}}_z(k) \\ & (\mathbf{1} - \widehat{\mathbf{s}}_z(k)) \circ \mathbf{s}_z(k) = \mathbf{0} \end{aligned} \tag{6}$$

The added equality constraints imply that the cancellation variables set in the first-level in the macroscopic sub-problem are kept in the mesoscopic one and the non-feasible short-turns allowed in the macroscopic sub-problem are kept in the mesoscopic one. Note that the symbol $\circ$ represents the element-wise product.

END SECOND LEVEL

Else solve the macroscopic MILP sub-problem for zone z'.

$$\begin{aligned} \min_{\mathbf{x}(k),\mathbf{v}'(k)} \quad & \mathbf{g}^{\mathrm{T}}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}'(k)_z}\mathbf{v}'(k)_z + \mathbf{g}^{\mathrm{T}}_{\mathbf{v}'(k)_{z'}}\mathbf{v}'(k)_{z'} \\ \text{s.t.} \quad & \mathbf{A}_{\mathbf{x}(k)}\mathbf{x}(k) + \mathbf{A}_{\mathbf{v}'(k)_z}\mathbf{v}'_z(k) + \mathbf{A}_{\mathbf{v}'_{z'}}\mathbf{v}'_{z'}(k)_{z'} \leq \mathbf{b}(k) \\ & \mathbf{v}'_{z'}(k) = \bar{\mathbf{v}}'_{z'}(k) \end{aligned} \tag{7}$$

Update n, Δ, and C_{time}; go to *P.2.2*.

Return the solution vector $[\widehat{\mathbf{x}}^{\mathrm{T}}(k)\ \ \widehat{\mathbf{v}}^{\mathrm{T}}_z(k)\ \ \widehat{\mathbf{v}}'^{\mathrm{T}}_{z'}(k)]$T where $\widehat{\mathbf{v}}^{\mathrm{T}}_z(k)$ is the solution vector of the mesoscopic problem for the disrupted zone and $\widehat{\mathbf{v}}'^{\mathrm{T}}_{z'}(k)$ is the solution vector of the macroscopic problem for all other zones.

Note that the above procedure can also be used for the case in which only small disturbances are present by setting the global macroscopic MILP problem as specified in Table II.

In the next section we describe the rescheduling algorithm that includes the above algorithm integrated in the MPC feedback control scheme.

V. The Rescheduling Algorithm

This section presents the proposed automatic on-line feedback control algorithm. Differently from the state of the art that mainly provides open-loop control techniques and looks at the global optimality instead of global feasibility of the solutions [15], the proposed rescheduling algorithm permits the on-line feedback control of the railway traffic and ensures the feasibility of the dispatching plans.

In particular, the algorithm allows to achieve the following three goals:

1) On-line feedback control of the railway traffic in the event of a full blockade and various short delays in a large-scale network. Note that the full blockade and short delays can occur at any time and that the nominal timetable can be cyclic or acyclic;
2) On-line computation of control actions that minimize the delays throughout the network as well as the cancellations of train runs and shunting actions;
3) On-line computation, in a rolling horizon mode, of feasible dispatching plans that allow the global reordering of the traffic in the whole network and the local management of the rolling stock in the disrupted stations.

For a detailed description of the fundamentals on MPC control the interested reader is referred to [33].

Algorithm 1 - Distributed optimization algorithm

Phase 1 – CONFIGURATION

P1.1 Global Macroscopic Problem Definition
Configure the global macroscopic MILP problem (2)

P1.2 Macroscopic Problem Partitioning
Partition (2) into α macroscopic MILP sub-problems (3)

P1.3 Mesoscopic Problem Definition
Set the mesoscopic MILP sub-problem (4) for the disrupted zone z

P1.4 Stop Criterion Definition
Define the algorithm stop criterion $(n \geq N_{max}) OR (\Delta \leq \Delta_{max}) OR (C_{time} \geq C_{max})$

Phase 2 – CORE ALGORITHM

P2.1 Set Counters - Set Initial Solution
Set the counter of the zones $z = 1$ and the iteration counter $n = 1$
Set the initial solution $\bar{x}(k) = [d^T_{nom}(k) a^T_{nom}(k)]^T$ and $\bar{v}'(k) = 0_{mx1}$

P2.2 Distributed Optimization
Iterate until $(n \geq N_{max}) OR (\Delta \leq \Delta_{max}) OR (C_{time} \geq C_{max})$
Iterate until $z \leq \alpha$
P2.2.1 Check Zone
if z is the disrupted zone **then**
execute the BI-LEVEL HEURISTICS
FIRST-LEVEL
Solve the macroscopic MILP sub-problem (5) for zone z
SECOND-LEVEL
Solve the mesoscopic MILP sub-problem (6) for zone z
else
solve the macroscopic MILP sub-problem (7) for zone z
end if

Return solution vector $[\hat{x}^T(k)\hat{v}^T_z(k)\hat{\mathbf{v}}'^T_{z'}(k)]^T$

Fig. 5. The pseudocode of the distributed rescheduling algorithm.

A. *The structure of the MPC algorithm*

The algorithm is sketched in Fig. 6 and consists of 4 steps. Step 1 is an offline configuration of the algorithm; Steps 2 to 4 perform the on-line feedback control of the system and are iteratively executed in accordance with the MPC scheme.

Step 1 - *Configuration*

In this preliminary step, the historical data regarding the

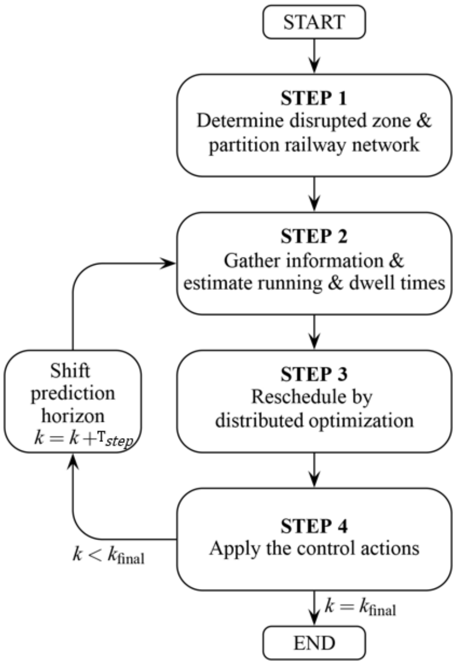

Fig. 6. The integrated rescheduling algorithm

unpredictable events that may affect the railway network are analyzed and the most common interruptions are identified and characterized in terms of duration and location in the railway network. Consequently, based on the performed analysis and on the company requirements and constraints (e.g., refresh rate of the current system state, communication systems, etc.) the following input parameters of the rescheduling algorithm are set:

- *sampling time* T_{step}, i.e., the time step for the MPC procedure to update the railway state in the system model and compute the optimal control actions;
- *number of iterations* $k_{\text{final}} = T - T_{\text{curr}}/T_{\text{step}}$, where T_{curr} is the time at which the algorithm starts and T corresponds to the daily timetable duration. Note that $k = 1,\ 2, \ldots, k_{\text{final}}$ represents the counter of iterations and at each iteration the time step is updated accordingly $t(k) = k\ T_{\text{step}}$;
- *estimated duration of the disruption* T_{disr};
- *number of partitioning zones* α, i.e., the number of zones in which the whole network would be partitioned for the implementation of the distributed optimization;
- *prediction horizon* T_{hor} for the MPC procedure, which should be higher than or equal to the estimated duration of the disruption;
- *control horizon* T_{contr};
- *transition period* T_{tr}, i.e., the time necessary to restore the nominal traffic after the disruption;
- *objective of the rescheduling problem.*

Based on the above parameters, the global MILP problems, both for the cases of small disturbances and small disturbances and disruption, are set and **Phase 1** of the integrated algorithm is executed.

Step 2 – *Update State*
At each time step $t(k)$ the current state of the system is measured (e.g., via the European Rail Traffic Management System) and the actual running times and dwell times are collected.

Step 3 – *Solve the Rescheduling Problem*
The state of the railway traffic is evaluated and only if delays, or a disruption or both are occurring, or if the system is in the transition phase (i.e., nominal traffic restoration after the disruption) the global rescheduling MILP problem is updated and solved over the prediction horizon T_{hor}.

Step 4 – *Control*
In this step, the control actions computed in Step 3 are applied for the current time step T_{step} and the prediction horizon is shifted of one time step, coherently with the MPC scheme. Note that if the railway traffic is in nominal conditions, no control action is applied.

After executing Step 4 the iteration counter k is incremented by one and the steps from 2 to 4 are iterated until the counter k is lower than the assigned number of iterations k_{final}.

We now focus on Step 3, which is the fundamental part of the algorithm and is summarized in Fig. 7. In particular, at each time step of the MPC procedure it is necessary to verify the state of the network, which can be in one of these four modes: (0) *Nominal schedule*, (1) *Delay*, (2) *Disruption*, and (3) *Transition* (i.e., the phase of nominal traffic restoring after the end of the disruption). The flowchart in Fig. 7 shows the actions performed for every iteration by the control technique and is described hereafter.

In *mode* (0), i.e., *Nominal schedule*, no control action must be performed and the nominal timetable is applied without any modification.

In *mode* (1), i.e., *Delay*, the network is affected by small delays. In this case, no cancellations, short-turn, shunting, nor ordering actions are necessary. The rescheduled timetable for T_{hor} is obtained by executing *Phase 2* of the integrated algorithm presented in Section IV considering the Macroscopic MILP model for the small disturbances case. Note that no disruption is occurring, hence the BI-LEVEL HEURISTICS is not executed.

In *mode* (2), i.e., *Disruption*, the network is affected by disruption and the rescheduled timetable for the prediction horizon is obtained by executing *Phase 2* of the integrated algorithm presented in Section IV considering the macroscopic MILP model for the small disturbances and disruption case. In this mode the BI-LEVEL HEURISTICS is executed for the disrupted zone.

Finally, in *mode* (3), i.e., *Transition*, the network is in a transition period immediately after the end of the disruption, where canceling, short-turning, and shunting actions are still necessary to finally restore the nominal functioning. In this situation, the rescheduled timetable for the prediction horizon is obtained by executing *Phase 2* of the integrated algorithm presented in Section IV considering the macroscopic MILP model for small disturbances and the disruption case.

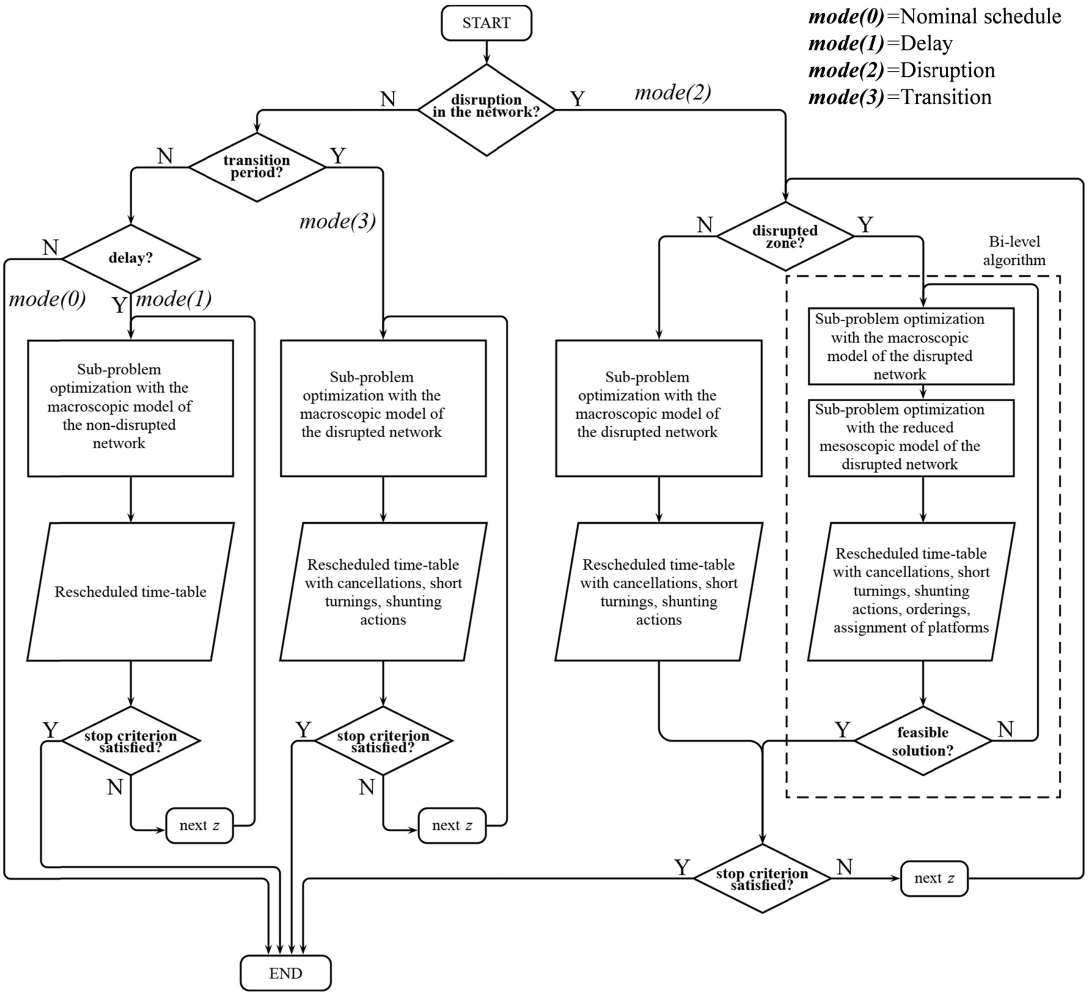

Fig. 7. Distributed optimization in one time step of the MPC procedure.

VI. Case Study and Simulation Results

In this section, the proposed rescheduling algorithm is used to reschedule the railway traffic of the national Dutch railway network in case of disruptions and short delays (Fig. 8). A full blockade of the country network is considered on the track section between the stations Lage Zwaluwe (LZW) and Dordrecht (DD). The considered section includes the Moerdijk bridge, at which disruptions often occur due to adverse weather conditions. Moreover, this section of the network is part of one of three important train routes from North to South of the Netherlands.

Figure 8 shows the main lines of the Dutch network and the zones in which the network is partitioned for the rescheduling (respectively represented in green, red, black, and blue colors). The disrupted area is represented in yellow in the green zone and the full blockade is indicated by an orange cross on the disrupted section. During the blockade, trains arriving from the south at LZW are short-turned and return to their starting destination. Similarly, trains arriving from the north at DD are short-turned and passengers continue their trip to LZW by bus. The short-turns of trains at stations DD and LZW may lead to local deviations from the nominal timetable that can cause secondary delays for the rest of the network.

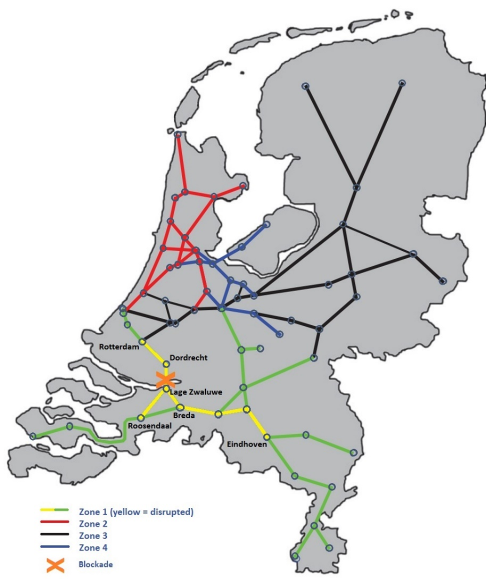

Fig. 8. The Dutch railway network.

The rescheduling algorithm is implemented in the Matlab (ver. R2020a) environment on an Intel core i7 processor with 8 Gb memory, and the optimization problem is solved with the Gurobi solver (ver. 7.0.2).

As performance indicators for the rescheduling algorithm we consider:

1) AvgAD, i.e., the average arrival delay;
2) MaxAD, i.e., the maximum arrival delay;
3) %DTR, i.e., the percentage of delayed train runs;
4) AvgCT, i.e., the average computation time per iteration in the MPC scheme;
5) StdAD, i.e., the standard deviation of the arrival delays;
6) StdCT, i.e., the standard deviation of the computation time per iteration in the MPC scheme.

The approach proposed in this paper is applied to five scenarios considering randomly generated delays in the network and a disruption between Dordrecht and Lage Zwaluwe with five different durations.

Scenario 1: $T_{\rm disr} = 60$ min, $T_{\rm hor} = 75$ min (i.e., longer than the duration of the disruption and sufficiently limited to avoid computation time issues).

Scenario 2: $T_{\rm disr} = 20$ min, $T_{\rm hor} = 75$ min.

Scenario 3: $T_{\rm disr} = 30$ min, $T_{\rm hor} = 75$ min.

Scenario 4: $T_{\rm disr} = 40$ min, $T_{\rm hor} = 75$ min.

Scenario 5: $T_{\rm disr} = 50$ min, $T_{\rm hor} = 75$ min.

For all scenarios we perform 20 short delay instances: Scenario1.1 to Scenario 1.20; Scenario 2.1 to Scenario 2.20; Scenario3.1 to Scenario 3.20; Scenario 4.1 to Scenario 4.20; Scenario5.1 to Scenario 5.20 considering that 20% of the train runs is delayed by unexpected short delays that follow a uniform distribution on the interval [1, 15] minutes, with 20 different seeds. Summarizing, we perform 100 different scenarios simulations to properly evaluate the proposed rescheduling algorithm.

The input parameters for Step 1 – *Configuration* of the rescheduling algorithm are set as follows:

- *sampling time* $T_{\rm step} = 1$ min;
- *number of iterations* $k_{\rm final} = 1440 - T_{\rm curr}$, where $T_{\rm curr}$ is the time at which the algorithm starts the control and 1440 corresponds to the daily timetable duration (in minutes);
- *estimated duration of the disruption* $T_{\rm disr}$ set depending on the considered Scenario;
- *number of zones* for the partitioning of the network $\alpha = 4$, depicted in Fig. 8 as the red, the blue, the black, and the green zone;
- *prediction horizon* $T_{\rm disr} \leq T_{\rm hor} \leq 120$ min;
- *control horizon* $T_{\rm contr}$=$T_{\rm hor}$;
- *transition period* $T_{\rm tr} = 30$ min;

Note that the transition period is set based on the suggestions by the train dispatchers.

The results obtained for the performance indicators are represented by means of boxplots, where in each blue box the central red mark indicates the median value, and the bottom and top edges of the blue box indicate the 25th and 75th percentiles, respectively. Whiskers extend to the most extreme data points that are not considered outliers, and outliers are plotted individually using the '+' symbol. The boxplots are reported in Fig. 9, while the corresponding minimum, maximum, median, 25th percentile, and 75th percentile values for each performance indicator and per each scenario are collected in Table III. The outcomes are as follows:

- AvgAD index (i.e., the average arrival delay) varies in the interval [3.87, 5.73] minutes for Scenarios from 1.1 to 1.20; [3.88, 5.49] minutes for Scenarios from 2.1 to 2.20; [3.90, 5.54] minutes for Scenarios from 3.1 to 3.20; [4.01, 5.65] minutes for Scenarios from 4.1 to 4.20; and [4.15, 4.79] minutes for Scenarios from 5.1 to 5.20. As shown by the boxplots and by the corresponding data reported in Table III, the algorithm performs best in Scenario 2, where the duration of the disruption is the shortest one, i.e., 20 minutes. In effect, analyzing the results in Table III the AvgAD presents the lowest values both for the 25th and the 75th percentile of the instances in Scenario 2. Nevertheless, it can be observed that in all scenarios the AvgAD in the 75% of the total delay instances is lower than 5.4 minutes. Consequently, it can be asserted that the algorithm is not particularly influenced by the duration of the disruption and in the considered scenarios it allows ensuring AvgAD $\approx 4.5 \pm 1$ minutes;
- MaxAD index (i.e., the maximum arrival delay) varies in the interval [14.24, 22.34] minutes for Scenarios from 1.1 to 1.20; [14.24, 26.33] minutes for Scenarios from 2.1 to 2.20; [14.28, 27.84] minutes for Scenarios from 3.1 to 3.20; [14.27, 27.84] minutes for Scenarios from 4.1 to 4.20, and [14.28, 27.84] minutes for Scenarios from 5.1 to 5.20. Analyzing the corresponding boxplots and the results reported in Table III, it can be observed that the value of the MaxAD index for the 25th percentile of the instances is lower than 16.55 minutes for all scenarios. Moreover, it can be observed that for the 75th percentile of the instances, the value of the MaxAD is lower than 27.85 minutes for all scenarios. As for the AvgAD index, the MaxAD values are not particularly influenced by the duration of the disruption. Furthermore, it is noticeable that the maximum arrival delay for all of the instances is much lower than the maximum admissible delay beyond which customers can request a partial or total refund of the ticket (i.e., 30 minutes for a partial refund and 60 minutes for a total refund).
- %DTR index (i.e., the percentage of delayed train runs) varies in the interval [5.21, 9.52] for Scenarios from 1.1 to 1.20; [5.28, 9.33] for Scenarios from 2.1 to 2.20; [5.26, 9.04] for Scenarios from 3.1 to 3.20; [5.20, 9.03] for Scenarios from 4.1 to 4.20; and [5.20, 9.09] for Scenarios from 5.1 to 5.20. Analyzing the corresponding boxplots and the results reported in Table III it can be observed that the value of the %DTR index for the 25th percentile of the instances is lower than 6.60%. Moreover, it can be observed that for the 75th percentile of the instances, the value of the %DTR is lower than 8.06% for all scenarios. Thus, it can be observed that the percentage of delayed train runs is particularly low although the network is affected both by a disruption and delays.
- AvgCT index (i.e., the average computation time per

iteration in the MPC scheme) varies in the interval $[26.43, 40.70]$ seconds for Scenarios from 1.1 to 1.20; $[25.24, 35.93]$ seconds for Scenarios from 2.1 to 2.20; $[22.93, 37.42]$ seconds for Scenarios from 3.1 to 3.20; $[20.87, 40.62]$ seconds for Scenarios from 4.1 to 4.20; and $[25.55, 54.60]$ seconds for Scenarios 5.1 to 5.20. Analyzing the corresponding boxplots and the results reported in Table III, it can be observed that the value of the AvgCT index for the 25th percentile of the instances is lower than 29.38 seconds. Moreover, it can be observed that for the 75th percentile of the instances, the value of the AvgCT is lower than 41.33 seconds for all scenarios. It is worth noting that the AvgCT remains lower than one minute (i.e., the time step of the MPC scheme) in all the considered instances while its variation depends on the disruption duration and short delay instance scenario.
- StdAD index (i.e., the standard deviation of the arrival delays) varies in the interval $[1.21, 2.23]$ minutes for Scenarios from 1.1 to 1.20; $[1.25, 2.07]$ minutes for Scenarios from 2.1 to 2.20; $[1.35, 2.16]$ minutes for Scenarios from 3.1 to 3.20; $[1.36, 2.16]$ minutes for Scenarios from 4.1 to 4.20; and $[1.36, 2.16]$ minutes for Scenarios from 5.1 to 5.20. Analyzing the corresponding boxplots and the results reported in Table III, it can be observed that the value of the StdAD index for the 25th percentile of the instances is lower than 1.49 minutes. Moreover, it can be observed that for the 75th percentile of the instances, the value of the StdAD is lower than 1.74 minutes for all scenarios. This shows that the variation of the arrival delays with respect to the average value is particularly limited in all scenarios although the disruption duration varies from 20 to 60 minutes.
- StdCT (i.e., the standard deviation of the computation time per iteration in the MPC scheme) varies in the interval $[17.30, 71.11]$ seconds for Scenarios from 1.1 to 1.20; $[16.54, 35.96]$ seconds for Scenarios from 2.1 to 2.20; $[16.54, 35.96]$ seconds for Scenario 3.1 to 3.20; $[13.89, 49.34]$ seconds for Scenarios from 4.1 to 4.20; and $[21.89, 56.66]$ seconds for Scenarios from 5.1 to 5.20. Analyzing the corresponding boxplots and the results reported in Table III, it can be observed that the value of the StdCT index for the 25th percentile of the instances is lower than 25.06 seconds. Moreover, it can be observed that for the 75th percentile of the instances, the value of the StdCT is lower than 35.87 seconds for all scenarios. Consequently, the values obtained for the standard deviation of the computation time confirm that the proposed rescheduling algorithm allows the resolution of the rescheduling problem for the considered large-scale network in less than one minute in all the implemented scenarios.

Note that for each time step the global MILP problem has 11 501 variables and 123 773 constraints for the worst-case Scenario 1; while it has 10 559 variables and 113 356 constraints for the best-case Scenario 2, which thus presents a lower complexity.

With the aim of evaluating the effectiveness of the rescheduling algorithm not only in managing railway traffic but also in properly handling the rolling stock during the disruption, in Fig. 10 we graphically report the assignment of trains to the available platforms at the disrupted stations. The two graphs in Fig. 10 respectively show the assignment of the trains to the platforms in LZW and DD stations in the most critical Scenario 1, i.e., when the duration of the disruption is equal to 60 minutes. It can be observed that in both stations the algorithm properly assigns the trains to the available platforms, respectively 4 in LZW and 6 in DD, without violating the corresponding capacity constraints and avoiding accidents due to the overlapping of dwell times on the same platforms. This confirms the effectiveness of the algorithm in properly fulfilling the rolling stock constraints in a large-scale systems affected by a severe disruption and various delays.

To further assess the outcomes of the proposed algorithm, we report the results obtained with a semi-heuristic algorithm that mimics the traditional control actions that can be performed in the two disrupted stations by a train dispatcher during the disruption. In particular, the heuristics considers a manual short-turning that combines each incoming train run with the first available outgoing train run, i.e., without any optimization. For the sake of brevity, we only report the results obtained for the worst-case and best-case scenarios with variable short delays, i.e., S1.1 to S1.20 and S2.1 to S2.20. In particular, the AvgAD index increases between 10% and 30%; the MaxAD index increases between 10% and 35%; the %DTR index increases between 35% and 50%; on the contrary the AvgCT index decreases between 20% and 30% with respect to the MPC approach. Consequently, the outcomes show that the choice of the heuristic traditional procedure for the assignment and ordering of train runs in the disrupted stations reasonably leads to the worsening of the arrival delays in the network and to an obvious reduction of the computation time necessary for the execution of the rescheduling algorithm, as compared to the performance of our algorithm.

Finally, the proposed MPC-based rescheduling algorithm has been further tested in Scenario 1 considering the exact resolution of the mesoscopic problem for the disrupted zone, thus excluding the bi-level heuristics. On average the algorithm requires 20 minutes for each iteration of the MPC scheme, which is far higher than the expected 1 minute, thus confirming the suitability of the proposed algorithm to be applied in a real-time feedback control environment.

Concluding, we want to highlight that the obtained results are already very good in terms of applicability to real-time Centralized Traffic Control, but they can be obviously further improved considering a dedicated high-performance computer for the implementation of the algorithm.

VII. Conclusions and Future Work

In this paper, we propose an innovative on-line feedback control algorithm for the rescheduling of railway traffic in case of a disruption and various delays in a large-scale network. Our method, based on the knowledge of the state

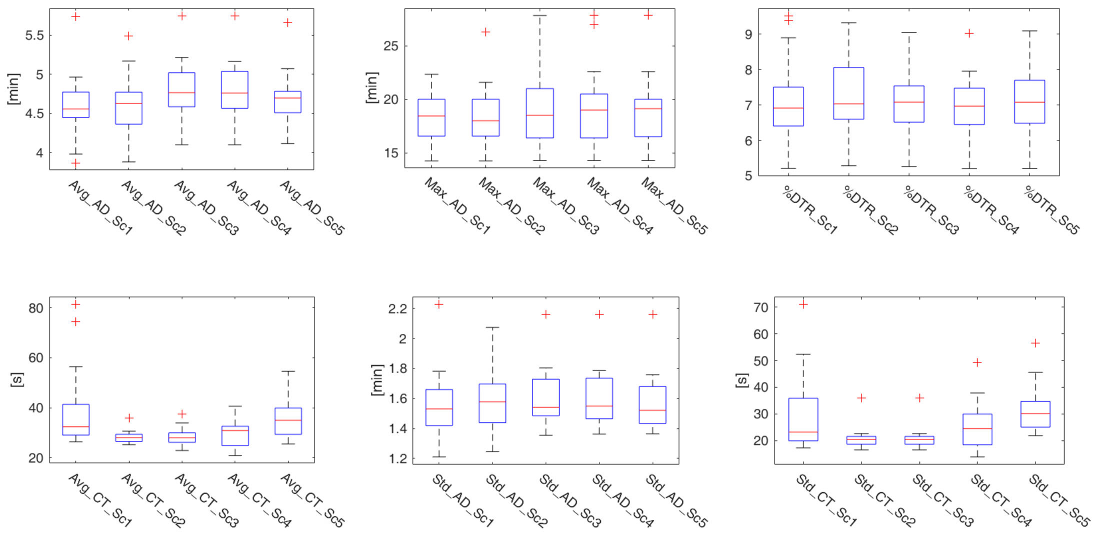

Fig. 9. Boxplots of the performance indices for the 20 instances of Scenarios S1, S2, S3, S4 and S5: AvgAD, MaxAD; %DTR, AvgCT, StdAD, StdCT.

TABLE III
BOXPLOT VALUES OF THE PERFORMANCE INDICATORS FOR SCENARIO 1.1 TO SCENARIO 5.20

Performance indicators		Scenario				
		Sc. 1.1 to 1.20	Sc. 2.1 to 2.20	Sc. 3.1 to 3.20	Sc. 4.1 to 4.20	Sc. 5.1 to 5.20
Avg AD [min]	Min	3.87	3.88	3.90	4.01	4.15
	Median	4.56	4.63	4.77	4.76	4.70
	Max	5.73	5.49	5.54	5.65	4.79
	25%	4.45	4.37	4.79	4.57	4.51
	75%	4.77	4.77	5.02	5.04	4.78
Max AD [min]	Min	14.24	14.24	14.28	14.27	14.28
	Median	18.44	18.00	18.50	19.00	19.13
	Max	22.34	26.33	27.84	27.84	27.84
	25%	16.55	16.55	16.39	16.39	16.51
	75%	20.00	20.00	21.00	20.50	20.00
% DTR	Min	5.21	5.28	5.26	5.20	5.20
	Median	6.92	7.03	7.08	6.97	7.08
	Max	9.52	9.33	9.04	9.03	9.09
	25%	6.41	6.60	6.52	6.45	6.48
	75%	7.51	8.06	7.54	7.48	7.70
Avg CT [s]	Min	26.43	25.24	22.93	20.87	25.55
	Median	32.40	28.08	28.02	30.87	34.99
	Max	40.75	35.93	37.92	40.62	54.60
	25%	29.07	26.53	26.19	24.89	29.38
	75%	41.33	29.41	29.98	32.63	39.89
Std AD [min]	Min	1.21	1.25	1.35	1.36	1.36
	Median	1.53	1.58	1.54	1.55	1.52
	Max	2.23	2.07	2.16	2.16	2.16
	25%	1.41	1.44	1.49	1.46	1.43
	75%	1.66	1.70	1.73	1.74	1.68
Std CT [s]	Min	17.30	16.54	16.54	13.89	21.89
	Median	23.22	20.46	20.46	24.48	30.16
	Max	71.11	35.96	35.96	49.34	56.66
	25%	19.89	18.56	18.66	18.42	25.06
	75%	35.87	21.56	21.56	29.96	34.72

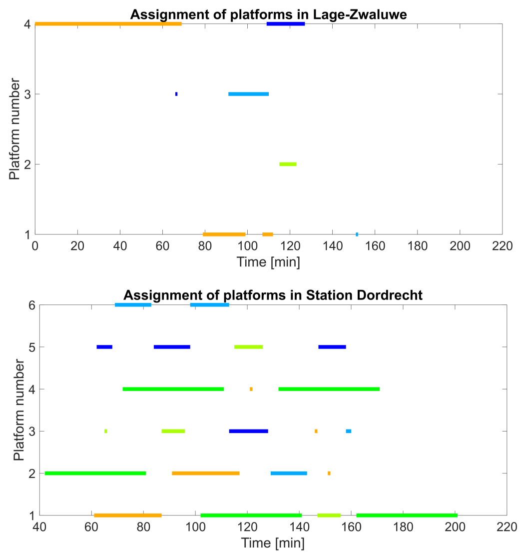

Fig. 10. Assignment of trains to platforms in the disrupted stations: Lage Zwaluwe and Dordrecht.

of the network, provides dynamically in a rolling horizon control mode a feasible rescheduled timetable that includes the physical operations to be performed in the disrupted stations. The proposed technique is based both on a macroscopic and a mesoscopic mixed integer linear programming model, and combines the Model Predictive Control (MPC) approach with an integrated algorithm that merges a distributed optimization method with a bi-level heuristics.

The integrated algorithm allows to reduce the computation time with respect to more classical algorithms that look at the optimality of the timetable but fail in providing feasible solutions for large-scale systems. The method has been tested on various disruption and delays scenarios for the real national Dutch railway network. The obtained outcomes highlight the effectiveness of the approach in minimizing the average and maximum arrival delays, the percentage of the delayed train runs, and the computation time, thus ensuring the time step constraint of 1 minute for the MPC approach (which is coherent with classical train dispatching specifications) without neglecting the physical limitations and the actual operations to be performed in the disrupted stations.

Further research will explore the effect of different tunings of the objective function, the robustness of the technique with respect to multiple disruptions and uncertainty in the data, and the energy consumption optimization.

References

[1] UIC - Sustainable Development Department, *Carbon Footprint of Railway Infrastructure*. International Union of Railways (UIC), 2016.

[2] R. Larsen, M. Pranzo, A. D'Ariano, F. Corman, and D. Pacciarelli, "Susceptibility of optimal train schedules to stochastic disturbances of process times," *Flex. Serv. Manuf. J.*, vol. 26, no. 4, pp. 466–489, Dec. 2014.

[3] L. Chen, C. Roberts, F. Schmid, and E. Stewart, "Modeling and solving real-time train rescheduling problems in railway bottleneck sections," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1896–1904, Aug. 2015.

[4] V. Cacchiani *et al.*, "An overview of recovery models and algorithms for real-time railway rescheduling," *Transportation Research Part B: Methodological*, vol. 63. Pergamon, pp. 15–37, May-2014.

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

[6] G. Cavone, M. Dotoli, N. Epicoco, and C. Seatzu, "A decision making procedure for robust train rescheduling based on mixed integer linear programming and Data Envelopment Analysis," *Appl. Math. Model.*, vol. 52, pp. 255–273, Dec. 2017.

[7] N. Ghaemi, O. Cats, and R. M. P. Goverde, "Railway disruption management challenges and possible solution directions," *Public Transp.*, vol. 9, no. 1–2, pp. 343–364, 2017.

[8] F. Chu and A. Oetting, "Modeling capacity consumption considering disruption program characteristics and the transition phase to steady operations during disruptions," *J. Rail Transp. Plan. Manag.*, vol. 3, no. 3, pp. 54–67, 2013.

[9] S. Shen and N. H. M. Wilson, "An Optimal Integrated Real-time Disruption Control Model for Rail Transit Systems," Springer, Berlin, Heidelberg, 2001, pp. 335–363.

[10] N. Ghaemi, O. Cats, and R. M. P. Goverde, "Macroscopic multiple-station short-turning model in case of complete railway blockages," *Transp. Res. Part C Emerg. Technol.*, vol. 89, pp. 113–132, Apr. 2018.

[11] 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.

[12] I. Louwerse and D. Huisman, "Adjusting a railway timetable in case of partial or complete blockades," *Eur. J. Oper. Res.*, vol. 235, no. 3, pp. 583–593, Jun. 2014.

[13] C. Hirai, T. Kunimatsu, N. Tomii, S. Kondou, and M. Takaba, "A Train Stop Deployment Planning Algorithm Using a Petri-net-based Modelling Approach," *Q. Rep. RTRI*, vol. 50, no. 1, pp. 8–13, 2009.

[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] F. Corman and L. Meng, "A review of online dynamic models and algorithms for railway traffic management," *IEEE Trans Intell Transp Sys*, vol. 16, no. 3, pp. 1274–1284, 2015.

[16] I. Louwerse and D. Huisman, "Adjusting a railway timetable in case of partial or complete blockades," *Eur. J. Oper. Res.*, vol. 235, no. 3, pp. 583–593, Jun. 2014.

[17] L. P. Veelenturf, M. P. Kidd, V. Cacchiani, L. G. Kroon, and P. Toth, "Large-scale disruptions a railway timetable rescheduling approach for handling large-scale disruptions," *Transp. Sci.*, vol. 50, pp. 1–24, Aug. 2015.

[18] J. Jespersen-Groth *et al.*, "Disruption management in passenger railway transportation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5868 LNCS, pp. 399–421.

[19] L. Meng and X. Zhou, "Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach," *Transp. Res. Part B Methodol.*, vol. 45, no. 7, pp. 1080–1102, Aug. 2011.

[20] N. Ghaemi, O. Cats, and R. M. P. Goverde, "Macroscopic multiple-station short-turning model in case of complete railway blockages," *Transp. Res. Part C Emerg. Technol.*, vol. 89, pp. 113–132, 2018.

[21] Y. C. Lai, S. W. Wang, and K. L. Huang, "Optimized Train-Set Rostering Plan for Taiwan High-Speed Rail," *IEEE Trans. Autom. Sci. Eng.*, 2017.

[22] A. Giua and C. Seatzu, "Modeling and Supervisory Control of Railway Networks Using Petri Nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 3, pp. 431–445, Jul. 2008.

[23] C. Hirai, T. Kunimatsu, N. Tomii, S. Kondou, and M. Takaba, "A train stop deployment planning algorithm using a Petri-net-based modelling approach," *Q. Rep. RTRI*, vol. 50, no. 1, pp. 8–13, 2009.

[24] 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.

[25] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice-A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[26] S. Lee, S. Kumara, and N. Gautam, "Market-based model predictive control for large-scale information networks: Completion time and value of solution," *IEEE Trans. Autom. Sci. Eng.*, 2008.

[27] 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.

[28] J. Rudan, B. Kersbergen, T. van den Boom, and K. Hangos, "Performance analysis of MILP based model predictive control algorithms for dynamic railway scheduling," in *2013 European Control Conference, ECC 2013*, 2013, pp. 4562–4567.

[29] G. Caimi, M. Fuchsberger, M. Laumanns, and M. Lüthi, "A model predictive control approach for discrete-time rescheduling in complex central railway station areas," *Comput. Oper. Res.*, vol. 39, no. 11, pp. 2578–2593, Nov. 2012.

[30] S. Li, B. De Schutter, L. Yang, and Z. Gao, "Robust Model Predictive Control for Train Regulation in Underground Railway Transportation," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 1075–1083, May 2016.

[31] B. Moaveni and S. Najafi, "Metro Traffic Modeling and Regulation in Loop Lines Using a Robust Model Predictive Controller to Improve Passenger Satisfaction," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1541–1551, Sep. 2018.

[32] L. Blenkers, "Railway disruption management," MSc final thesis. Faculty 3mE, Delft University of Technology, 2015.

[33] J. H. Lee, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation and Systems*. 2011.