Delft University of Technology

Delft Center for Systems and Control

Technical report 15-002

Towards railway traffic management using switching max-plus-linear systems – Structure analysis and rescheduling*

B. Kersbergen, J. Rudan, T. van den Boom, and B. De Schutter

If you want to cite this report, please use the following reference instead:

B. Kersbergen, J. Rudan, T. van den Boom, and B. De Schutter, "Towards railway traffic management using switching max-plus-linear systems – Structure analysis and rescheduling," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 26, no. 2, pp. 183–223, 2016.

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`

*This report can also be downloaded via `https://pub.deschutter.info/abs/15_002.html`

Towards Railway Traffic Management Using Switching Max-Plus-Linear Systems

Structure Analysis and Rescheduling

Bart Kersbergen · János Rudan · Ton van den Boom · Bart De Schutter

Abstract In this paper we present a railway traffic model and a model predictive controller for online railway traffic management of railway networks with a periodic timetable. The main aim of the controller is to recover from delays in an optimal way by changing the departure of trains, by breaking connections, by splitting joined trains, and - in the case of multiple tracks between two stations - by redistributing the trains over the tracks. The railway system is described by a switching max-plus-linear model. We assume that measurements of current running and dwell times and estimates of future running times and dwell times are continuously available so that they can be taken into account in the optimization of the system's control variables. The switching max-plus-linear model railway model is used to determine optimal dispatching actions, based on the prediction of the future arrival and departure times of the trains, by recasting the dispatching problem as a Mixed Integer Linear Programming (MILP) problem and solving it. Moreover, we use properties from max-plus algebra to rewrite and reduce the model such that the MILP problem can be solved in less time. We also apply the algorithm to a model of the Dutch railway network.

Keywords Max-plus algebra · Railway networks · Rescheduling · Model reduction · MILP

1 Introduction

In recent years a lot of research effort has been oriented towards the design of timetables that are robust against propagation of delays in the railway network,

B. Kersbergen · T.J.J. van den Boom · B. De Schutter
Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628CD Delft, The Netherlands
E-mail: {B.Kersbergen, A.J.J.vandenBoom, B.DeSchutter}@tudelft.nl

J. Rudan
Faculty of Information Technology, Pázmány Péter Catholic University, 1083 Budapest, Práter utca 50/a ,Budapest, Hungary
E-mail: rudan.janos@itk.ppke.hu

caused by technical failures, fluctuation of passenger volumes, and weather influence (Goverde 2007, 2010; Kroon and Peeters 2003). Developing decision support systems for dispatchers, that determine new conflict-free schedules and routes for the railway traffic when delays occurs, has been a topic of interest for many researchers in recent years (Caimi et al 2012; Corman et al 2012; Kecman et al 2013; Törnquist Krasemann 2012).

In Caimi et al (2012) a decision support system is developed that tries to schedule and route all trains in an area in and around a large station. A model predictive control approach with a microscopic model of the railway operations is used based on blocking times. At each point where rescheduling of a train is possible, a set of possible blocking times for different routes and departure times at platforms or arrival times at the boundaries of the area are considered for that train. As many trains as possible are then assigned a route with corresponding blocking times while all safety and operational constraints are respected. The objective is to optimize the passenger satisfaction, measured by punctuality and reliability. The resulting optimization problem is a binary linear programming problem. Our approach differs in the scale of the networks considered and the detail of the model. We aim to schedule the railway traffic for the entire national railway network and to be able to do that we model the railway traffic and network as a macroscopic model. We do not consider routing problems of trains through station areas.

In D'Ariano et al (2007) the railway operation is also modeled as a microscopic model based on blocking times with an Alternative Graph approach. In their alternative graph approach for every train occupying a block section a node is created in the graph. The nodes of a single train are then connected to each other through running time constraints and for every pair of trains occupying the same block section headway/separation constraints are added. If the order in which the trains can occupy the block sections can be changed with dispatching actions, then a pair of alternative arcs defining the two orders in which the trains can occupy the block section are added to the graph. A new schedule for the railway traffic is found when for each pair of alternative arcs only one arc is chosen. The graph has an extra node, to which all nodes are connected and the weights of the arcs from all nodes to this extra node are chosen such that minimizing the maximum weight of all paths from the starting to the ending node corresponds to minimizing the maximum consecutive delay. To solve this problem the authors use their own branch and bound algorithm. Corman et al (2012) has extended the work of D'Ariano et al (2007) to also consider breaking connections. For different sets of maintained connections the maximum consecutive delay is determined and the decrease of the maximum consecutive delay is weighted against the number of broken connections. The biggest differences in modeling between this work and our work is the level of detail considered and the solver used to solve the problem. We do not consider block sections, but only tracks between stations and the interlocking area of a station is considered as a single node. Our models are also built as cyclic models allowing us to easily expand the simulation and control period to multiple cycles without having to rebuild the entire model. The method for solving the optimization problem is also different. D'Ariano et al (2007) and Corman et al (2012) use a specifically designed branch and bound algorithm made for minimizing the maximum consecutive delay. We use state of the art Mixed Integer Linear Programming (MILP) solvers allowing us to freely choose our objective function,

as long as it is a linear function. This makes it easier to add dispatching actions such as breaking connections and changing tracks, and incorporate a penalty for these actions in the cost function.

In Kecman et al (2013) the same Alternative Graph approach as in Corman et al (2012) and D'Ariano et al (2007) is used but now for macroscopic models with different levels of details. The level of detail in this model is similar to our own work. One of the differences between their work and ours is that they model the system as an alternative graph and not as a max-plus-linear system and the alternative graph is specifically designed such that the maximum consecutive delay can be minimized using the branch and bound algorithm of D'Ariano et al (2007). This branch and bound algorithm is specifically designed to minimize the weight of the longest path and the alternative graph is adjusted such that the weight of the longest path corresponds to the maximum consecutive delays. If they want to consider breaking connections they have to adjust the whole solution procedure as was seen in Corman et al (2012). In our approach the cost of breaking connections, and other dispatching actions can be added by simply adjusting the cost function by changing the weights on the decision variables corresponding to those actions.

Most of the previously mentioned approaches use a microscopic model. The advantage of a microscopic model is that the railway operation is modeled in more detail and is therefore more accurate. The railway operations in stations are also modeled in full detail; all tracks, platforms, signals, and switches are considered. With these details the decision support system can determine the optimal routes and schedule for the trains at the stations as well. The downside to using a microscopic model is that the increased detail and accuracy result in a very complex model. Only the railway operation of a part of the whole network is considered when solving the dispatching problem, since it would take too much time to find the optimal solution to the dispatching problem for the entire network, to be of use in an on-line dispatching support system. The authors of Törnquist Krasemann (2012) limit the computation time needed to solve the dispatching problem by using a greedy method to find a good solution as fast as possible, and improve on this solution if time permits it, instead of trying to solve the problem to optimality. Next we will describe our approach to the decision support system in general

In our approach the decision support system is divided into four subsystems:

- A monitoring system
- A model predictive controller
- The model of the model predictive controller
- A route planning system

In Figure 1 it is shown how these systems are connected to each other and to the actual railway operations. Currently there is no system for the tracking of all trains in the Netherlands in rail-time and this is the case in many other countries as well. As a result the monitoring of the trains in The Netherlands has to be done based on data from the signaling system. Once the data from the signaling system has been processed the model predictive controller uses the data to update the model and determine the optimal dispatching actions based on the current state of the network and the predictions of the effects of the dispatching actions. The result is an updated timetable that is conflict free with respect to the considered model. Due to the changes in the timetable, and the reordering of trains, some

trains need new routes. These routes and trajectories are determined in the route planning system. Although it does not show in the figure, there is also feedback from the route planning system to the model when the route planning system cannot find feasible routes for all trains. The model predictive controller then adjusts its model and process times and recomputes a new timetable. The route planning then determines new routes for the new timetable. The new timetable and routes are then given to the dispatchers who can implement them on the railway system.

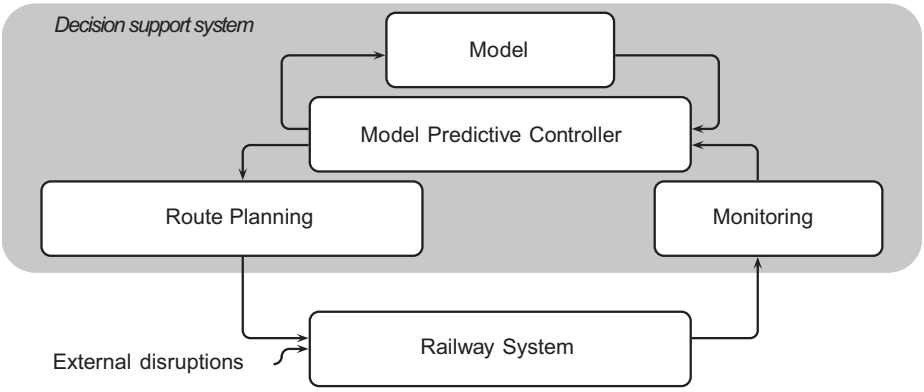

Fig. 1 The decision support system

In this paper we concentrate on the model predictive controller and the model used to predict the future arrival and departure times of the trains. Particularly we focus on how the model is built and on improving the time needed to solve a single step of the model predictive controller. In a single step the model predictive controller solves the dispatching problem for the railway operations on the entire passenger railway network using a macroscopic model of the railway operations for a given prediction and control horizon. The dispatching actions are limited to changing the order of trains, breaking train connections, changing the tracks trains are driving on, and breaking joined trains. Rerouting trains in and around the interlocking areas of stations is not considered. We focus on a single step because each step needs to be solved very fast, since the controller needs to be usable during real-time operations. Because the problem is an MILP problem the computation time will, in the worst case, increase exponentially with the number of binary control variables. If we want to be able to use the system to help dispatchers solve the rescheduling problem for the network of a whole country for a long prediction and control horizon we need to be able solve the problem fast, especially when several iterations may be necessary when the route planning cannot find feasible routes.

In Braker (1991, 1993), de Vries et al (1998), Heidergott and Vries (2001), de Waal et al (1997), and Minciardi et al (1995) it has been shown that a model of a railway network, with a fixed routing schedule and fixed connections, can be described as a max-plus-linear model. In max-plus algebra such a model is

'linear'. Max-plus algebra uses maximization and addition as its basic operations (Baccelli et al 1992). A system that can be described by a max-plus-linear model can be characterized as a discrete event system in which only synchronization occurs, but no concurrency or choice (Baccelli et al 1992). However, in the case of delays, it is sometimes better to make small changes to the schedule and possibly break connections. By taking these actions the total delay in the network may be reduced, at the cost of some extra delays for a small part of the trains. The current paper builds on the work of de Vries et al (1998) and Heidergott and Vries (2001), where the concept of controlling the connections of train in a railway network using max-plus algebra is introduced. This paper also builds on the work of van den Boom and De Schutter (2006), van den Boom et al (2011), van den Boom et al (2012), and Kersbergen et al (2013a,b), where a controlled railway system is modeled using *switching* max-plus-linear (SMPL) models. In this description a number of max-plus-linear models is used, where each model corresponds to a specific mode, describing the railway network by a different set of connection constraints, a different train schedule and a different routing plan. The system is controlled by switching between different modes, allowing us to break train connections, to change the order of trains, and to change the track the trains run over in case there are multiple tracks to choose from between stations or junctions. In De Schutter et al (2002) it has been shown that model predictive control is a suitable option for reducing the delays in railway networks. We continue the work of van den Boom et al (2012) by extending the analysis of the structure of the system matrices with the introduction of the coupling matrix and separate headway matrices for the headway constraints between arrivals and departures. The mathematical notation is also improved in such a way that we no longer need the $\odot$-operator, used for the max-plus Schur product, and we make use of max-plus binary variables, which is more in line with the notation of Kersbergen et al (2013a,b). The new notation ensures the model can be described within the max-plus algebra, which in turn ensures that the conversion of the model into its explicit form is possible, since the conversion of the model from its implicit to its explicit form is based on a theory from max-plus algebra and that is only valid for max-plus models. Furthermore, we extend the results of Kersbergen et al (2013a,b) by showing how the matrix structure can be used to calculate the explicit model. We also extend the computational results of Kersbergen et al (2013a,b) with a new case study where we measure the computation time for solving the MILP problems for different cost functions.

In Section 2 the nominal operation of the railway traffic on the railway network is described, converted to a max-plus-linear (MPL) model, and the structure of the system matrices is analyzed. In Section 3 the model is extended so the train orders can be changed, connections can be broken, joined trains can be split up, and trains can change track if there are multiple tracks available. The effects of this extension on the model structure is also analyzed in this section. In Section 4 it is explained how the model of the previous section can be converted into its explicit form and how the model structure of the system matrices can be used to convert it. In Section 5 the model is reduced by limiting the control freedom. In Section 6 a case study is performed on a model of a railway network for different objective functions of the rescheduling problem. In Section 7 conclusions are drawn and recommendations for future research are made.

2 Nominal Operation

Consider a railway network for a passenger trains, where the trains are operating according to a periodic railway timetable with a period T. This railway network consists of a set of tracks and a set of stations. Each track starts and ends at a station and trains cannot overtake each other on the tracks. In this paper we generalize the concept of a 'station' to places where the train order may be changed. This includes actual stations and their interlocking areas, but also junctions outside the interlocking areas.

2.1 Model

The nominal operation of the railway network is modeled as a cyclic discrete-event system, with cycle counter k. The events of the discrete-event system are the arrival and departure events of the trains at the stations. In one cycle all departure events of all trains for one period of the timetable, and their corresponding arrival events, are modeled. For all past cycles $k - n$, $n \in \mathbb{N}$, all event times are known, fixed, and in the past.

The combination of the following actions: a train departing from a station, traversing a track, and arriving at the next station, is called a train run. Each train run has an index i, and an associated departure time d_i and arrival time a_i. A set of train runs, modeling the same 'physical' train, will be called a line. During nominal operation, the railway traffic operates according to the nominal timetable: the trains follow their pre-determined routes, the order in which the trains depart and arrive at stations is fixed, all connections are maintained and there are no delays in the network. The operation of the railway network can be described as a set of train runs connected to each other through various constraints.

2.2 Constraints Connecting the Train Runs

There are six different constraints connecting the trains; these are:

- Running time constraints
- Continuity constraints
- Timetable constraints
- Headway constraints
- Coupling constraints
- Connection constraints

Next all of these constraints are described in more detail.

2.2.1 Running time constraints

The relation between the arrival time and departure time of a train run can be described by a *running time* constraint. In the rest of this paper we will simply refer to a 'train run' as a 'train'. A running time constraint is defined such that the arrival and departure of a train belong to the same cycle.

This results in the following definition for a running time constraint for train i in cycle k:

$$a_i(k) \geq d_i(k) + \tau_{\mathrm{r},i}(k), \tag{1}$$

where $\tau_{\mathrm{r},i}(k)$ is the *running time*, i.e. the time the train needs to traverse the track, for train run i in cycle k. Process times may vary each cycle, even during nominal operation, e.g. due to a changing number of passengers and different rolling stock, and therefore the running times depend on k.

2.2.2 Continuity constraints

A continuity constraint connects two trains of the same line to each other. For example, a 'physical' train driving from one station to the next and then continuing on to a third station. This can be modeled by considering train i and its predecessor p_i. Let train p_i model the 'physical' train driving from the first to the second station and let train i model the 'physical' train driving from the second to the third station. Train i can then only start some time after train p_i has arrived:

$$d_i(k) \geq a_{p_i}(k - \mu_{i,p_i}) + \tau_{\mathrm{d},i,p_i}(k), \tag{2}$$

where $\mu_{i,p_i} = 0$ if train p_i in cycle k continues as train i in cycle k and $\mu_{i,p_i} = \alpha$ if train p_i in cycle $k - \alpha$ continues as train i in cycle k, and $\tau_{\mathrm{d},i,p_i}(k)$ is the *dwell time*, i.e. the time the train waits at the station for passengers to board and alight.

2.2.3 Timetable constraints

Since the passenger railways operate according to a timetable, none of the trains are allowed to depart before their scheduled departure times and in some cases they may not arrive before their scheduled arrival times either. This requirement can be modeled by adding *timetable* constraints:

$$d_i(k) \geq r_{\mathrm{d},i}(k) \tag{3}$$

$$a_i(k) \geq r_{\mathrm{a},i}(k), \tag{4}$$

where $r_{\mathrm{d},i}(k)$ and $r_{\mathrm{a},i}(k)$ are the scheduled departure and arrival time of train i in cycle k. Note that for a cyclic timetable it holds that $r_{\mathrm{d},i}(k) = r_{\mathrm{d},i}(0) + kT$ and $r_{\mathrm{a},i}(k) = r_{\mathrm{a},i}(0) + kT$. In many countries trains are allowed to arrive before their scheduled arrival time; in that case the timetable constraint on the arrival time, as in (4), should be left out for those trains.

2.2.4 Headway constraints

Headway constraints define the order in which trains traverse tracks and they indirectly define the minimum distance between trains. This is done by relating the arrival and departure times of one train to the arrival and departure time of the other trains traversing the same track. The headway times are chosen such that, as long as there are no unexpected delays on the tracks, none of the trains run into a yellow or red signal and have to break. If several trains traverse a track in the same direction, then for train i the set $\mathcal{H}_i$ is defined as the set of trains that start on the track before train i and traverse the track in the same direction during

nominal operation. The headway constraints for train i for the trains traversing the track in the same direction are:

$$d_i(k) \geq d_l(k - \mu_{i,l}) + \tau_{\mathrm{h,d},i,l}(k) \tag{5}$$
$$a_i(k) \geq a_l(k - \mu_{i,l}) + \tau_{\mathrm{h,a},i,l}(k), \tag{6}$$

for each $l \in \mathcal{H}_i$, where $\tau_{\mathrm{h,d},i,l}(k)$ is the *headway time* for departures, i.e. the time needed between the departure of train l in cycle $k - \mu_{i,l}$ and the departure of train i in cycle k, $\tau_{\mathrm{h,a},i,l}(k)$ is the headway time needed between the arrival of train l in cycle $k - \mu_{i,l}$ and the arrival of train i in cycle k, and where $\mu_{i,l}$ is defined in the same way as for (2).

If trains traverse the track in the opposite direction, then for train run i, the set $\mathcal{S}_i$ is defined as the set of trains that start on the same track before train i and traverse the track in opposite direction during nominal operation. The headway constraints for train i for the trains traversing the track in the opposite direction are:

$$d_i(k) \geq a_m(k - \mu_{i,m}) + \tau_{\mathrm{s},i,m}(k), \tag{7}$$

for each $m \in \mathcal{S}_i$, where $\tau_{\mathrm{s},i,m}(k)$ is the *separation time*, i.e. the time the train i in cycle k must wait before it can enter the track after train m in cycle $k - \mu_{i,m}$ has left the track and where $\mu_{i,m}$ is defined in the same way as for (2).

2.2.5 Coupling constraints

At some stations two 'physical' trains are coupled and continue as a single 'physical' train; this is modeled by *coupling* constraints. The coupling constraints ensure that the arrival and departure times of the two 'physical' trains are the same. Consider train i and let train o_i be the train to which train i should be coupled to during nominal operation. For these trains the coupling constraints are:

$$d_i(k) = d_{o_i}(k) \tag{8}$$
$$a_i(k) = a_{o_i}(k). \tag{9}$$

Here it is assumed both trains are in the same cycle, since they have the same departure and arrival times. These two equality constraints can be written as four inequality constraints:

$$d_i(k) \geq d_{o_i}(k) \tag{10}$$
$$d_{o_i}(k) \geq d_i(k) \tag{11}$$
$$a_i(k) \geq a_{o_i}(k) \tag{12}$$
$$a_{o_i}(k) \geq a_i(k). \tag{13}$$

These four inequality constraints are used instead of two equality constraints because in the perturbed operation (see Section 3) it will be possible to cancel the coupling if one of the trains is delayed and using these inequalities instead of equalities makes it easier to support that modification.

2.2.6 Connection constraints

At some stations passengers can transfer to another train. Transfers that are guaranteed, are modeled by *connection* constraints. Connection constraints ensure that passengers can change trains at stations by defining a relation between the departure time of one train and the arrival time of the train from which the passengers transfer. Define $\mathcal{C}_i$ as the set of train runs, train i has to give a connection to during nominal operation. Then the connection constraints for train i are defined as:

$$d_i(k) \geq a_e(k - \mu_{i,e}) + \tau_{\mathrm{c},i,e}(k), \tag{14}$$

for each $e \in \mathcal{C}_i$, and where $\tau_{\mathrm{c},i,e}(k)$ is the *connection time*, i.e. the time needed for the passengers to transfer from train e in cycle $k - \mu_{i,e}$ to train i in cycle k.

2.3 Max-Plus Algebra

In the next subsection we will use max-plus algebra to describe the model of the railway traffic for nominal operation. Therefore we will introduce the basic concepts of max-plus algebra in this subsection.

The max-plus algebra is an idempotent semi-ring, consisting of the set $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$, where $\varepsilon = -\infty$, equipped with the two operators $\oplus$ and $\otimes$, that are defined as follows (Baccelli et al 1992; Cuninghame-Green 1979; Heidergott et al 2006):

$$\begin{aligned} a \oplus b &= \max(a, b) \\ a \otimes b &= a + b, \end{aligned}$$

for $a, b \in \mathbb{R}_\varepsilon$. During evaluation $\otimes$ has priority over $\oplus$.

For matrices these operators are defined as:

$$\begin{aligned} [A \oplus B]_{i,j} &= [A]_{i,j} \oplus [B]_{i,j} = \max([A]_{i,j}, [B]_{i,j}) \\ [A \otimes C]_{i,j} &= \bigoplus_{w=1}^{n} [A]_{i,m} \otimes [C]_{m,j} = \max_{w=1,\ldots,n} ([A]_{i,m} + [C]_{m,j}), \end{aligned}$$

where $A, B \in \mathbb{R}_\varepsilon^{m \times n}$ and $C \in \mathbb{R}_\varepsilon^{n \times p}$.

The matrix $\mathcal{E}$ is the max-plus-algebraic zero matrix: $\mathcal{E}_{ij} = \varepsilon$ for all i, j. A max-plus diagonal matrix $\mathcal{D} = \mathrm{diag}_\oplus(\delta_1, \ldots, \delta_n) \in \mathbb{R}_\varepsilon^{n \times n}$ has elements $[\mathcal{D}]_{i,j} = \varepsilon$ for $i \neq j$ and diagonal elements $[\mathcal{D}]_{i,i} = \delta_i$ for $i = 1, \ldots, n$. A max-plus permutation matrix $T \in \mathbb{R}_\varepsilon^{m \times m}$ has one zero in each row and one zero in each column and ε elsewhere.

Furthermore, define a max-plus binary control variable as $u \in \{0, \varepsilon\}$ and define the adjoint $\overline{u}$ as:

$$\overline{u} = \begin{cases} \varepsilon & \text{if } u = 0 \\ 0 & \text{if } u = \varepsilon, \end{cases} \tag{15}$$

which will be used in Section 3 and the sections there after.

2.4 Max-Plus-Linear Model

If all constraints are satisfied, the trains can depart and arrive without running into any conflicts with other trains. Therefore, we assume that all trains depart and arrive as soon as all constraints are satisfied. Then the constraints for train i can be written using two equations, one for the arrival and one for the departure time:

$$\begin{aligned} d_i(k) = \max\Bigg(& a_{p_i}(k-\mu_{i,p_i}) + \tau_{\mathrm{d},i,\mathrm{p}_i}(k), \\ & \max_{l\in\mathcal{H}_i}\Big(d_l(k-\mu_{i,l}) + \tau_{\mathrm{h,d},i,l}(k)\Big), \\ & \max_{m\in\mathcal{S}_i}\Big(a_m(k-\mu_{i,m}) + \tau_{\mathrm{s},i,m}(k)\Big), \\ & \max_{e\in\mathcal{C}_i}\Big(a_e(k-\mu_{i,e}) + \tau_{\mathrm{c},i,e}(k)\Big), \\ & d_{o_i}(k),\, r_{\mathrm{d},i}(k)\Bigg) \end{aligned} \tag{16}$$

$$\begin{aligned} a_i(k) = \max\Bigg(& \max_{l\in\mathcal{H}_i}\Big(a_l(k-\mu_{i,l}) + \tau_{\mathrm{h,a},i,l}(k)\Big), \\ & d_i(k) + \tau_{\mathrm{r},i}(k),\, a_{o_i}(k),\, r_{\mathrm{a},i}(k)\Bigg). \end{aligned} \tag{17}$$

Note that in an undisturbed, well-defined time schedule the terms $r_{\mathrm{d},i}(k)$ and $r_{\mathrm{a},i}(k)$ in (16) and (17) respectively will be the largest. However, if one of the trains p_i, l, m, e or o_i has a delay, due to unforeseen circumstances (an incident, a late departure, etc.), then the corresponding term can become larger than the others and train i will depart later than the scheduled departure time $r_{\mathrm{d},i}(k)$ and will therefore be delayed as well.

Now let us consider a network with n 'trains' and define the vectors

$$x(k) = \begin{bmatrix} d_1(k) \\ \vdots \\ d_n(k) \\ a_1(k) \\ \vdots \\ a_n(k) \end{bmatrix} \in \mathbb{R}_\varepsilon^{2n}, \qquad r(k) = \begin{bmatrix} r_{\mathrm{d},1}(k) \\ \vdots \\ r_{\mathrm{d},n}(k) \\ r_{\mathrm{a},1}(k) \\ \vdots \\ r_{\mathrm{a},n}(k) \end{bmatrix} \in \mathbb{R}_\varepsilon^{2n}.$$

By defining appropriate matrices $A_\mu(k) \in \mathbb{R}_\varepsilon^{2n\times 2n}$ for $\mu = 0, 1, \ldots, \mu_{\max}$, where $\mu_{\max} = \max_{i,j} \mu_{i,j}$, (16) and (17) can be rewritten as

$$\begin{aligned} x_i(k) = \max\big(& \max_j(x_j(k) + [A_0(k)]_{i,j}), \max_j(x_j(k-1) + [A_1(k)]_{i,j}), \ldots, \\ & \max_j(x_j(k-\mu_{\max}) + [A_{\mu_{\max}}(k)]_{i,j}), r_i(k)\big), \end{aligned} \tag{18}$$

where $[A_\mu(k)]_{i,j}$ is the (i,j)th entry of[1] $A_\mu(k)$.

[1] The matrices $A_\mu(k)$ can be completed by adding $[A_\mu(k)]_{i,j} = -\infty$ for all combinations (μ, i, j, k) that do not appear in (18).

Using the max-plus algebra, (18) can be written as a max-plus-linear equation:

$$x_i(k) = r_i(k) \oplus \bigoplus_{\mu=0}^{\mu_{\max}} \bigoplus_{j=1}^{2n} x_j(k-\mu) \otimes [A_\mu(k)]_{i,j}. \tag{19}$$

By determining this max-plus-linear equation for all $x_i(k)$ the model can be written as a max-plus-linear (MPL) model defined as:

$$x(k) = r(k) \oplus \bigoplus_{\mu=0}^{\mu_{\max}} A_\mu(k) \otimes x(k-\mu), \tag{20}$$

which is an MPL model of a railway network with a fixed routing schedule and fixed connections, such as the models of Braker (1991) and Goverde (2010). In the next sections we will omit (k) from the notation of the $A_\mu(k)$ matrices to improve the readability.

2.5 System Matrices for the Nominal Operation

From (20) it is clear that the dynamics of the railway system are described by the matrices A_μ, $\mu = 0, \ldots, \mu_{\max}$. In this section we will study the structure of these matrices for the nominal operation. It can be verified that the matrices A_μ, $\mu = 0, \ldots, \mu_{\max}$ can be written as

$$A_\mu = \begin{bmatrix} A_{\mu,4,\mathrm{d}} \oplus A_{\mu,6,\mathrm{d}} & A_{\mu,2} \oplus A_{\mu,3} \oplus A_{\mu,5} \\ A_{\mu,1} & A_{\mu,4,\mathrm{a}} \oplus A_{\mu,6,\mathrm{a}} \end{bmatrix}, \tag{21}$$

with $A_{\mu,1}$, $A_{\mu,2}$, $A_{\mu,3}$, $A_{\mu,4,\mathrm{d}}$, $A_{\mu,4,\mathrm{a}}$, $A_{\mu,5}$, $A_{\mu,6,\mathrm{d}}$, $A_{\mu,6,\mathrm{a}} \in \mathbb{R}_\varepsilon^{n \times n}$. The structure of these six matrices will now be discussed in detail.

2.5.1 The running time matrix

The running time matrix, denoted by $A_{\mu,1}$, represents the running time constraints. Since by definition the arrival and departure of a train are in the same cycle, $A_{\mu,1} = \mathcal{E}$ for all $\mu \neq 0$. As a result, $A_{\mu,1}$ has the following structure:

$$A_{\mu,1} = \begin{cases} \mathrm{diag}_\oplus(\tau_{\mathrm{r},1}(k), \tau_{\mathrm{r},2}(k), \cdots, \tau_{\mathrm{r},n}(k)) & \text{for } \mu = 0 \\ \mathcal{E} & \text{for } \mu \neq 0. \end{cases} \tag{22}$$

2.5.2 The dwell time matrix

The dwell time matrix, denoted by $A_{\mu,2}$, represents the continuity constraints. The structure of this matrix is defined as follows:

$$[A_{\mu,2}]_{i,j} = \begin{cases} \tau_{\mathrm{d},i,p_i}(k) & \text{if } j = p_i \text{ and } \mu = \mu_{i,p_i} \\ \varepsilon & \text{else.} \end{cases} \tag{23}$$

Continuity constraints are only defined between trains of the same line that directly follow each other, i.e. between train i and its predecessor $p_i = j$.

Let n_{L} be the number of lines in the network, let $n_{\mathrm{l},m}$ be the number of trains on line m, $m = 1, \ldots, n_{\mathrm{L}}$, so $n_{\mathrm{l},1} + n_{\mathrm{l},2} + \ldots + n_{\mathrm{l},n_{\mathrm{L}}} = n$, and let $L_m \in \mathbb{R}^{n_{\mathrm{l},m}}$ be a vector containing the indices of the trains of line m, with $L_{m,i}$ the ith element of vector L_m. Define a max-plus permutation matrix $E_{\mathrm{dwell}} \in \mathbb{R}_{\varepsilon}^{n \times n}$ that orders the rows and columns of $A_{\mu,2}$, such that the associated event times are ordered by line[2] and per line by the departure times in the timetable:

$$A_{\mu,2} = E_{\mathrm{dwell}} \otimes \begin{bmatrix} \hat{A}_{\mu,2,1} & \mathcal{E} & \cdots & \mathcal{E} \\ \mathcal{E} & \hat{A}_{\mu,2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathcal{E} \\ \mathcal{E} & \cdots & \mathcal{E} & \hat{A}_{\mu,2,n_L} \end{bmatrix} \otimes E_{\mathrm{dwell}}^{\top}, \tag{24}$$

where $\hat{A}_{\mu,2,m} \in \mathbb{R}^{n_{\mathrm{l},m} \times n_{\mathrm{l},m}}$ can be described as:

$$\hat{A}_{\mu,2,m} = \begin{bmatrix} \varepsilon & \cdots & \cdots & \varepsilon & \hat{\tau}_{\mathrm{d},m,\mu,1,n_{\mathrm{l},m}}(k) \\ \hat{\tau}_{\mathrm{d},m,\mu,2,1}(k) & \ddots & \ddots & \vdots & \varepsilon \\ \varepsilon & \hat{\tau}_{\mathrm{d},m,\mu,3,2}(k) & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \varepsilon & \vdots \\ \varepsilon & \cdots & \varepsilon & \hat{\tau}_{\mathrm{d},m,\mu,n_{\mathrm{l},m},n_{\mathrm{l},m}-1}(k) & \varepsilon \end{bmatrix}, \tag{25}$$

where

$$\hat{\tau}_{\mathrm{d},m,\mu,i,j}(k) = \begin{cases} \tau_{\mathrm{d},L_{m,i},L_{m,j}}(k) & \text{if } L_{m,j} = p_{L_{m,i}} \text{ and } \mu = \mu_{L_{m,i},L_{m,j}} \\ \varepsilon & \text{else.} \end{cases}$$

If $L_{m,i} = p_{L_{m,j}}$ then train $L_{m,i}$ is the predecessor of train $L_{m,j}$ – in other words, train $L_{m,i}$ continues as train $L_{m,j}$ – and there should be a dwell time between the arrival of train $L_{m,j}$ and the departure of train $L_{m,i}$.

2.5.3 The connection matrix

The connection matrix represents the connection constraints and is denoted by $A_{\mu,3}$. This matrix is structured as follows:

$$[A_{\mu,3}]_{i,j} = \begin{cases} \tau_{\mathrm{c},i,j}(k) & \text{if } j \in \mathcal{C}_i \text{ and } \mu = \mu_{i,j} \\ \varepsilon & \text{else.} \end{cases} \tag{26}$$

If $j \in \mathcal{C}_i$, then train i has to give a connection to train j; therefore a constraint is set on the departure time of train i which depends on the arrival time of train j.

[2] Recall that a line is defined as the set of train runs modeling the same 'physical' train.

2.5.4 The headway matrices

The matrices $A_{\mu,4,\mathrm{d}}$ and $A_{\mu,4,\mathrm{a}}$ represent the headway constraints for trains in the same direction on the same track.

Let n_{T} be the number of tracks in the network, let $n_{\mathrm{t},m}$ be the number of trains on track m, $m = 1, \ldots, n_{\mathrm{T}}$, so $n_{\mathrm{t},1} + n_{\mathrm{t},2} + \ldots + n_{\mathrm{t},n_{\mathrm{T}}} = n$, and let $T_m \in \mathbb{R}^{n_{\mathrm{t},m}}$ be a vector containing the indices of the trains on track m, ordered according to the timetable. Define a max-plus permutation matrix $E_{\mathrm{t}} \in \mathbb{R}_{\varepsilon}^{n \times n}$ that reorders the half of the state vector $x(k)$ containing the departure events $d_1(k)$ up to $d_n(k)$ such that the event times are ordered per track and for each track the events are ordered according to the timetable. The matrices $A_{\mu,4,\mathrm{d}}$ can then be defined as:

$$A_{\mu,4,\mathrm{d}} = E_{\mathrm{t}} \otimes \begin{bmatrix} \hat{A}_{\mu,4,\mathrm{d},1} & \mathcal{E} & \cdots & \mathcal{E} \\ \mathcal{E} & \hat{A}_{\mu,4,\mathrm{d},2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathcal{E} \\ \mathcal{E} & \cdots & \mathcal{E} & \hat{A}_{\mu,4,\mathrm{d},n_{\mathrm{T}}} \end{bmatrix} \otimes E_{\mathrm{t}}^{\top}, \tag{27}$$

where $\hat{A}_{\mu,4,\mathrm{d},m} \in \mathbb{R}^{n_{\mathrm{t},m} \times n_{\mathrm{t},m}}$ and where for $\mu = 0$ we have

$$\hat{A}_{0,4,\mathrm{d},m} = \begin{bmatrix} \varepsilon & \hat{\tau}_{\mathrm{h,d},m,0,1,2}(k) & \cdots & \hat{\tau}_{\mathrm{h,d},m,0,1,n_{\mathrm{t},m}}(k) \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \hat{\tau}_{\mathrm{h,d},m,0,n_{\mathrm{t},m}-1,n_{\mathrm{t},m}}(k) \\ \varepsilon & \cdots & \cdots & \varepsilon \end{bmatrix}, \tag{28}$$

and for $\mu = 1, \ldots, \mu_{\max}$

$$[\hat{A}_{\mu,4,\mathrm{d},m}]_{i,j} = \hat{\tau}_{\mathrm{h,d},m,\mu,i,j}(k), \tag{29}$$

with

$$\hat{\tau}_{\mathrm{h,d},m,\mu,i,j}(k) = \begin{cases} \tau_{\mathrm{h,d},T_{m,i},T_{m,j}}(k) & \text{if } T_{m,j} \in \mathcal{H}_{T_{m,i}}, \text{ and } \mu = \mu_{T_{m,i},T_{m,j}} \\ \varepsilon & \text{else.} \end{cases} \tag{30}$$

The if-condition is satisfied if the jth train in track m is in the set $\mathcal{H}_{T_{m,i}}$, which is the set of trains that traverse track m before the ith train on track m in the same direction. When it is satisfied there should be a headway time between the departure times of the two trains.

For $A_{\mu,4,\mathrm{a}}$, the structure of the matrix is the same as for $A_{\mu,4,\mathrm{d}}$, the only differences being that in (27)- (30) $\hat{A}_{\mu,4,\mathrm{d},m}$ is replaced by $\hat{A}_{\mu,4,\mathrm{a},m}$, $\hat{\tau}_{\mathrm{h,d},m,\mu,i,j}(k)$ is replaced by $\hat{\tau}_{\mathrm{h,a},m,\mu,i,j}(k)$, and $\tau_{\mathrm{h,d},T_{m,i},T_{m,j}}(k)$ is replaced by $\tau_{\mathrm{h,a},T_{m,i},T_{m,j}}(k)$.

2.5.5 The separation matrix

The separation matrix is denoted by $A_{\mu,5}$, and represents the headway constraints for trains driving over the same track in the opposite direction. Using the same

permutation matrix E_{t} as for the headway matrices, $A_{\mu,5}$ can be written as:

$$A_{\mu,5} = E_{\mathrm{t}} \otimes \begin{bmatrix} \hat{A}_{\mu,5,1} & \mathcal{E} & \cdots & \mathcal{E} \\ \mathcal{E} & \hat{A}_{\mu,5,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathcal{E} \\ \mathcal{E} & \cdots & \mathcal{E} & \hat{A}_{\mu,5,n_{\mathrm{T}}} \end{bmatrix} \otimes E_{\mathrm{t}}^{\top}, \tag{31}$$

where $\hat{A}_{\mu,5,m} \in \mathbb{R}^{n_{\mathrm{t},m} \times n_{\mathrm{t},m}}$ and can be described as:

$$[\hat{A}_{\mu,5,m}]_{i,j} = \begin{cases} \tau_{\mathrm{s},T_{m,i},T_{m,j}}(k) & \text{if } T_{m,j} \in \mathcal{S}_{T_{m,i}} \text{ and } \mu = \mu_{T_{m,i},T_{m,j}} \\ \varepsilon & \text{else.} \end{cases} \tag{32}$$

The if-condition is satisfied if the jth train in track m is in the set $\mathcal{S}_{T_{m,i}}$, which is the set of trains that traverse track m before the ith train on track m in the opposite direction. When the if-condition is satisfied there should be a separation time between the departure time of the ith train and the arrival time the jth train on track m.

2.5.6 The coupling matrices

The coupling matrices, denoted by $A_{\mu,6,\mathrm{d}}$ and $A_{\mu,6,\mathrm{a}}$, define which trains are coupled into one single train. The matrices $A_{\mu,6,\mathrm{d}}$ and $A_{\mu,6,\mathrm{a}}$ have the following structure:

$$[A_{\mu,6,\mathrm{d}}]_{i,j} = \begin{cases} 0 & \text{if } j = o_i \text{ and } \mu = 0 \\ 0 & \text{if } i = o_j \text{ and } \mu = 0 \\ \varepsilon & \text{else.} \end{cases} \tag{33}$$

Furthermore, $A_{\mu,6,\mathrm{a}} = A_{\mu,6,\mathrm{d}}$. The reason for defining two matrices that are identical is that during perturbed operation these matrices may differ: they will depend on control variables and different process times and for most combinations of control variables they will be different.

3 Perturbed Operation

The model, as described in the previous section, is a static model: the order of the trains on the tracks is fixed, connections cannot be broken, coupled trains cannot be decoupled, process times are fixed, and when there are multiple tracks trains can use, they cannot change tracks. In this section the abilities to change the order of the trains, break connections, decouple trains, and change tracks will be added step by step. These abilities are added by extending the model of the previous section to a switching max-plus-linear (SMPL) model by modifying constraints and adding max-plus binary control variables $u_{i,l}(k-\mu)$ and their adjoint $\overline{u_{i,l}(k-\mu)}$, as defined in (15), to the constraints. It is called an SMPL model, since it can switch between behaviors (train orders, track choices, broken connections). The SMPL model can be described as:

$$x(k) = r(k) \oplus \bigoplus_{\mu=0}^{\mu_{\max}} A_\mu(k, u(k-\mu)) \otimes x(k-\mu) \oplus \bigoplus_{\mu=-\mu_{\max}}^{-1} A_\mu(k, u(k)) \otimes x(k-\mu), \quad (34)$$

where the elements of A_μ contain the modified constraints with control variables $u_{i,l}(k-\mu)$ and $\overline{u_{i,l}(k-\mu)}$, and u is the control vector containing all control variables. The first max-plus sum for $\mu = 0, 1, \ldots, \mu_{\max}$ contains the part of the model that describes the dependency of the event times of the current cycle to the event times of previous cycles. Since the event times of previous cycles are known, fixed and in the past the control variables relating these event times to event times of the current cycle are also fixed, known and in the past. As a result A_μ depends on $u(k-\mu)$ when $\mu \geq 0$ and not on $u(k)$. By adding the ability to change the order of the trains, break connections, decouple trains, and change tracks new constraints are introduced between trains in the current cycle k and future cycles $k-\mu$, for $\mu < 0$. For example if a train in the current cycle is delayed a lot and is delaying a train in the next cycle we should be able to change the order of these trains. This involves constraints between an event in cycle k and in cycle $k+1$. By adding the second max-plus sum from $-\mu_{\max}$ to -1 these constraints can also be modeled. Since the control variables in the second max-plus sum are taken during the kth cycle, A_μ in the second max-plus sum depends on $u(k)$.

For $A_\mu(k, u(k-\mu))$, with $\mu \in \{0, \ldots, \mu_{\max}\}$, and their submatrices, the argument in the next part of the section is always $(k, u(k-\mu))$ and will therefore be omitted. For $A_\mu(k, u(k))$, with $\mu \in \{-\mu_{\max}, \ldots, -1\}$, and their submatrices, the argument is always $(k, u(k))$ and will also be omitted.

3.1 Changing the Order of Trains

To change the order of trains on a track, the headway constraints need to be manipulated. As an example consider two trains running over the same track and in the same direction. These trains are described by train i in cycle k and l in cycle $k-\mu_{i,l}$. If the order of trains is "l before i", then headway constraints (5) and (6) define this order. If the order is "i before l" then the following headway constraints should replace equations (5) and (6):

$$d_l(k-\mu_{i,l}) \geq d_i(k) \otimes \tau_{\mathrm{h,d},l,i}(k-\mu_{i,l}) \quad (35)$$

$$a_l(k-\mu_{i,l}) \geq a_i(k) \otimes \tau_{\mathrm{h,a},l,i}(k-\mu_{i,l}) \quad (36)$$

To be able to change the order of trains, it is necessary to be able to turn headway constraints on and off. This can be done by multiplying (5) and (6) with $u_{i,l}(k-\mu_{i,l})$, and (35) and (36) with $\overline{u_{i,l}(k-\mu_{i,l})}$. Then we obtain:

$$d_i(k) \geq d_l(k-\mu_{i,l}) \otimes \tau_{\mathrm{h,d},i,l}(k) \otimes u_{i,l}(k-\mu_{i,l}) \quad (37)$$

$$a_i(k) \geq a_l(k-\mu_{i,l}) \otimes \tau_{\mathrm{h,a},i,l}(k) \otimes u_{i,l}(k-\mu_{i,l}) \quad (38)$$

$$d_l(k-\mu_{i,l}) \geq d_i(k) \otimes \tau_{\mathrm{h,d},l,i}(k-\mu_{i,l}) \otimes \overline{u_{i,l}(k-\mu_{i,l})} \quad (39)$$

$$a_l(k-\mu_{i,l}) \geq a_i(k) \otimes \tau_{\mathrm{h,a},l,i}(k-\mu_{i,l}) \otimes \overline{u_{i,l}(k-\mu_{i,l})}. \quad (40)$$

A detailed example describing the effects of the control variables on (37)-(40) can be found in Appendix A.1.

For two trains running over the same track, and in opposite direction the same procedure of multiplying the constraints by control inputs can be applied:

$$d_i(k) \geq a_m(k-\mu_{i,m}) \otimes \tau_{\mathrm{s},i,m}(k) \otimes u_{i,m}(k-\mu_{i,m}) \tag{41}$$

$$d_m(k-\mu_{i,m}) \geq a_i(k) \otimes \tau_{\mathrm{s},m,i}(k-\mu_{i,m}) \otimes \overline{u_{i,m}(k-\mu_{i,m})} \tag{42}$$

Using this methodology the matrices $A_{\mu,4,\mathrm{d}}$, $A_{\mu,4,\mathrm{a}}$, and $A_{\mu,5}$ can be modified to allow reordering of trains. The new matrices can be described by (27) and (31) respectively, where $\hat{A}_{\mu,4,\mathrm{d},m} \in \mathbb{R}^{n_{\mathrm{t},m} \times n_{\mathrm{t},m}}$ and $\hat{A}_{\mu,5,m} \in \mathbb{R}^{n_{\mathrm{t},m} \times n_{\mathrm{t},m}}$ can be described as:

$$[\hat{A}_{\mu,4,\mathrm{d},m}]_{i,j} = \begin{cases} \tau_{\mathrm{h,d},T_{m,i},T_{m,j}}(k) \otimes u_{T_{m,i},T_{m,j}}(k-\mu) & \text{if } T_{m,j} \in \mathcal{H}_{T_{m,i}} \text{ and} \\ & \mu = \mu_{T_{m,i},T_{m,j}} \\ \tau_{\mathrm{h,d},T_{m,i},T_{m,j}}(k) \otimes \overline{u_{T_{m,j},T_{m,i}}(k)} & \text{if } T_{m,i} \in \mathcal{H}_{T_{m,j}} \text{ and} \\ & \mu = -\mu_{T_{m,j},T_{m,i}} \\ \varepsilon & \text{else,} \end{cases} \tag{43}$$

$$[\hat{A}_{\mu,5,m}]_{i,j} = \begin{cases} \tau_{\mathrm{s},T_{m,i},T_{m,j}}(k) \otimes u_{T_{m,i},T_{m,j}}(k-\mu) & \text{if } T_{m,j} \in \mathcal{S}_{T_{m,i}} \text{ and} \\ & \mu = \mu_{T_{m,i},T_{m,j}} \\ \tau_{\mathrm{s},T_{m,i},T_{m,j}}(k) \otimes \overline{u_{T_{m,j},T_{m,i}}(k)} & \text{if } T_{m,i} \in \mathcal{S}_{T_{m,j}} \text{ and} \\ & \mu = -\mu_{T_{m,j},T_{m,i}} \\ \varepsilon & \text{else,} \end{cases} \tag{44}$$

where T_m is again the vector containing the indices of the trains on track m, ordered according to the timetable. The first if-condition in both equations describes the nominal situation where the jth train on track m is in the set $\mathcal{H}_{T_{m,i}}$ or $\mathcal{S}_{T_{m,i}}$ respectively and denotes the headway or separation times between the trains with the added control variable $u_{T_{m,i},T_{m,j}}(k-\mu)$. The second if-condition in both equations states that if the default order of the trains on track m is the ith train before the jth train then element i,j should contain a headway or separation time respectively and an adjoint control variable $\overline{u_{T_{m,j},T_{m,i}}(k)}$, since it corresponds to a train order that is different from the nominal order.

For $A_{\mu,4,\mathrm{a}}$, the structure of the matrix is the same as for $A_{\mu,4,\mathrm{d}}$, the only differences are that in (43) $\hat{A}_{\mu,4,\mathrm{d},m}$ is replaced by $\hat{A}_{\mu,4,\mathrm{a},m}$, and $\tau_{\mathrm{h,d},T_{m,i},T_{m,j}}(k)$ is replaced by $\tau_{\mathrm{h,a},T_{m,i},T_{m,j}}(k)$.

The reader should note that for a track with n trains running over it the number of control variables added to the model is $n(n-1)/2$. The number of possible combinations of control actions is then $2^{n(n-1)/2}$, but the number of possible train order is only $n!$. For $n \geq 3$ there are more combinations of control variables than possible train orders. As a result some combinations of control variables do not correspond to possible train orders, they describe infeasible train orders. For an example of an infeasible train order see Appendix A.2, where we will also show that an infeasible train order results in infinite departure and arrival times of some trains.

The reason for adding more control variables than necessary is because it is easier to formulate the model: each combination of two trains on a track has one control variable that determines the order.

3.2 Breaking Connections

Breaking connections can be done by manipulating the entries of $A_{\mu,3}$. By setting the elements of $A_{\mu,3}$ to ε the connection is broken. This can be done by adding a control variable $u_{i,j+n}(k-\mu_{i,j}) \in \{\varepsilon, 0\}$ to the connection constraint, as was already shown in de Vries et al (1998) and Heidergott and Vries (2001). If $u_{i,j+n}(k-\mu_{i,j+n}) = \varepsilon$ the connection is broken, if $u_{i,j+n}(k-\mu_{i,j+n}) = 0$ the connection is maintained. This results in a new matrix:

$$[A_{\mu,3}]_{i,j} = \begin{cases} \tau_{\mathrm{c},i,j}(k) \otimes u_{i,j+n}(k-\mu_{i,j}) & \text{if } j \in \mathcal{C}_i \text{ and } \mu = \mu_{i,j} \\ \varepsilon & \text{else.} \end{cases} \tag{45}$$

The indices of the control variables are determined by the indices the element has in A_μ. Since $A_{\mu,3}$ is in the top right part of A_μ index i remains the same and index j is shifted by n to $j+n$. This also ensures there is no overlap in the indices of the control variables for the different dispatching actions. The indices of the connection times and μ are determined by the trains they are related to and since index j and $j+n$, for $j \leq n$, both correspond to train j (departure and arrival time), the indices do not need to be shifted.

3.3 Decoupling Trains

When two trains are decoupled, the coupling constraints need to be removed and headway constraints need to be added. These headway constraints should also allow the trains to depart in a different order. For the decoupling new max-plus binary variables are used: $v_{i,j}(k)$ and its adjoint $\overline{v_{i,j}(k)}$. These max-plus binary variables are defined as in (15). This results in the following build-up of the coupling matrix $A_{\mu,6,d}$:

$$[A_{\mu,6,\mathrm{d}}]_{i,j} = \begin{cases} 0 \otimes u_{i,j}(k) \oplus \tau_{\mathrm{h,d},i,j}(k) \otimes \overline{v_{i,j}(k)} \otimes u_{i,j}(k) & \text{if } j = o_i,\ i > j, \\ & \text{and } \mu = 0 \\ 0 \otimes u_{i,j}(k) \oplus \tau_{\mathrm{h,d},i,j}(k) \otimes \overline{v_{i,j}(k)} \otimes \overline{u_{i,j}(k)} & \text{if } i = o_j,\ i < j, \\ & \text{and } \mu = 0 \\ \varepsilon & \text{else,} \end{cases} \tag{46}$$

where $v_{i,j}(k)$ is the control variable for (de)coupling and $u_{i,j}(k)$ is the control variable that determines the order of the train departures if the trains are decoupled. The matrices $\hat{A}_{\mu,6,\mathrm{a}}$ are defined in the same way as $\hat{A}_{\mu,6,\mathrm{d}}$ with only one difference: $\tau_{\mathrm{h,d},i,j}(k)$ is replaced by $\tau_{\mathrm{h,a},i,j}(k)$.

Clearly if $v_{i,j}(k) = 0$ then $\overline{v_{i,j}(k)} = \varepsilon$ and the trains remain coupled. If $\overline{v_{i,j}(k)} = 0$ then either $[A_{\mu,6},\mathrm{d}]_{i,j} = \tau_{\mathrm{h,d},i,j}(k)$ and $[A_{\mu,6}]_{j,i} = \varepsilon$ or $[A_{\mu,6}]_{i,j} = \varepsilon$ and $[A_{\mu,6,\mathrm{d}}]_{j,i} = \tau_{\mathrm{h,d},j,i}(k)$.

3.4 Switching Between Tracks

Between certain stations in a railway network, there may be two (or more) parallel tracks that can be used by trains. To determine which track should be used, extra

control variables are added per train. In this paper we will be dealing with at most two parallel tracks in each direction (two sets of two unidirectional tracks). This is done for the sake of simplicity. As a result only one control variable has to be added per train traversing one of these parallel tracks. However, this approach can easily be generalized to the case where there are more parallel tracks. In that case one control variable per track per train is added, that indicates whether the train is on that track. Although this will result in many more control variables than needed in theory, it will keep the constraints as simple as possible.

The tracks are numbered in such a way that parallel tracks always have consecutive numbers. If trains can switch between track m and $m+1$, then this can be modeled by changing the headway matrices $A_{\mu,4,\mathrm{d}}$. Consider the part of $A_{\mu,4,\mathrm{d}}$, as described in (27), consisting of

$$\tilde{A}_{\mu,4,\mathrm{d},m} = \begin{bmatrix} \hat{A}_{\mu,4,\mathrm{d},m} & \mathcal{E} \\ \mathcal{E} & \hat{A}_{\mu,4,\mathrm{d},m+1} \end{bmatrix},$$

where $\hat{A}_{\mu,4,\mathrm{d},m}$ and $\hat{A}_{\mu,4,\mathrm{d},m+1}$ contain the headway times for the departures of the trains on track m and $m+1$, respectively. Define T_m as the vector containing the indices of the trains on track m, ordered according to the timetable and $\tilde{T}_m = [T_m^\top \, T_{m+1}^\top]^\top$ is the vector containing the indices of the trains on track m and track $m+1$. To enable switching between parallel tracks new max-plus binary variables are introduced: $w_{i,j}(k)$ and its adjoint $\overline{w_{i,j}(k)}$. These max-plus binary variables are defined as in (15). Switching between tracks can then be done by redefining the entries of $\tilde{A}_{\mu,4,\mathrm{d},m}$ as follows:

$$\left[\tilde{A}_{\mu,4,\mathrm{d},m}\right]_{i,j} = \begin{cases} \begin{array}{c} \tau_{\mathrm{h,d},\tilde{T}_{m,i},\tilde{T}_{m,j}}(k) \otimes \\ u_{\tilde{T}_{m,i},\tilde{T}_{m,j}}(k-\mu) \otimes \\ \big(w_{\tilde{T}_{m,i}}(k) \otimes w_{\tilde{T}_{m,j}}(k-\mu) \oplus \\ \overline{w_{\tilde{T}_{m,i}}(k)} \otimes \overline{w_{\tilde{T}_{m,j}}(k-\mu)}\big) \end{array} & \begin{array}{l} \text{if } \tilde{T}_{m,j} \in \mathcal{H}_{\tilde{T}_{m,i}} \text{ and} \\ \mu = \mu_{\tilde{T}_{m,i},\tilde{T}_{m,j}} \end{array} \\ \begin{array}{c} \tau_{\mathrm{h,d},\tilde{T}_{m,i},\tilde{T}_{m,j}}(k) \otimes \\ \overline{u_{\tilde{T}_{m,j},\tilde{T}_{m,i}}(k)} \otimes \\ \big(w_{\tilde{T}_{m,i}}(k) \otimes w_{\tilde{T}_{m,j}}(k-\mu) \oplus \\ \overline{w_{\tilde{T}_{m,i}}(k)} \otimes \overline{w_{\tilde{T}_{m,j}}(k-\mu)}\big) \end{array} & \begin{array}{l} \text{if } \tilde{T}_{m,i} \in \mathcal{H}_{\tilde{T}_{m,j}} \text{ and} \\ \mu = -\mu_{\tilde{T}_{m,j},\tilde{T}_{m,i}} \end{array} \\ \begin{array}{c} \tau_{\mathrm{h,d},\tilde{T}_{m,i},\tilde{T}_{m,j}}(k) \otimes \\ u_{\tilde{T}_{m,i},\tilde{T}_{m,j}}(k-\mu) \otimes \\ \big(w_{\tilde{T}_{m,i}}(k) \otimes \overline{w_{\tilde{T}_{m,j}}(k-\mu)} \oplus \\ \overline{w_{\tilde{T}_{m,i}}(k)} \otimes w_{\tilde{T}_{m,j}}(k-\mu)\big) \end{array} & \begin{array}{l} \text{if } r_{\mathrm{d},\tilde{T}_{m,i}}(k) \geq r_{\mathrm{d},\tilde{T}_{m,j}}(k-\mu), \\ \mu = \mu_{\tilde{T}_{m,i},\tilde{T}_{m,j}} \geq 0 \text{ and} \\ \tilde{T}_{m,i} \in \mathcal{T}_m,\ \tilde{T}_{m,j} \in \mathcal{T}_{m+1} \text{ or} \\ \tilde{T}_{m,i} \in \mathcal{T}_m,\ \tilde{T}_{m,j} \in \mathcal{T}_m \end{array} \\ \begin{array}{c} \tau_{\mathrm{h,d},\tilde{T}_{m,i},\tilde{T}_{m,j}}(k) \otimes \\ \overline{u_{\tilde{T}_{m,j},\tilde{T}_{m,i}}(k)} \otimes \\ \big(w_{\tilde{T}_{m,i}}(k) \otimes \overline{w_{\tilde{T}_{m,j}}(k-\mu)} \oplus \\ \overline{w_{\tilde{T}_{m,i}}(k)} \otimes w_{\tilde{T}_{m,j}}(k-\mu)\big) \end{array} & \begin{array}{l} \text{if } r_{\mathrm{d},\tilde{T}_{m,j}}(k-\mu) > r_{\mathrm{d},\tilde{T}_{m,i}}(k), \\ \mu = -\mu_{\tilde{T}_{m,j},\tilde{T}_{m,i}} \leq 0 \text{ and} \\ \tilde{T}_{m,i} \in \mathcal{T}_m,\ \tilde{T}_{m,j} \in \mathcal{T}_{m+1} \text{ or} \\ \tilde{T}_{m,i} \in \mathcal{T}_{m+1},\ \tilde{T}_{m,j} \in \mathcal{T}_m. \end{array} \end{cases} \tag{47}$$

The control variables $u_{\tilde{T}_{m,i},\tilde{T}_{m,j}}(k)$, $u_{\tilde{T}_{m,i},\tilde{T}_{m,j}}(k-\mu)$ are used to determine the order of the trains $\tilde{T}_{m,i}$ and $\tilde{T}_{m,j}$. The control variables $w_{\tilde{T}_{m,i}}(k)$ and $w_{\tilde{T}_{m,j}}(k-\mu)$

are used to determine on which track trains $\tilde{T}_{m,i}$ and $\tilde{T}_{m,j}$ are respectively. The values are chosen such that if $w_{\tilde{T}_{m,j}}(k) = 0$ and $w_{\tilde{T}_{m,j}}(k-\mu) = 0$, trains $\tilde{T}_{m,i}$ and $\tilde{T}_{m,j}$ are on the same track as in the nominal case. The first two if-statements describe the headway constraints for trains that are on the same track during nominal operation with the added control variables for the different tracks. The last two if-statements are the headway constraints between trains, that during the nominal operation, traverse parallel tracks. The default order between these trains is chosen according to their scheduled departure times.

For $\tilde{A}_{\mu,4,\mathrm{a},m}$, the structure of the matrix is the same as for $\tilde{A}_{\mu,4,\mathrm{d},m}$, the only differences is that in (47) $\tau_{\mathrm{h,d},\tilde{T}_{m,i},\tilde{T}_{m,j}}(k)$ is replaced by $\tau_{\mathrm{h,a},\tilde{T}_{m,i},\tilde{T}_{m,j}}(k)$.

An example of the constraints for four trains running over two parallel tracks can be found in Appendix A.3.

4 Explicit Model

The model introduced in (34) has a specific structure called the implicit form. In an equation in the implicit form the state vector $x(k)$ does not only depend on the state vector of the previous cycles (and the timetable reference), but also on itself. There also exist models where the state vector $x(k)$ does not depend on itself, only on the state vector of the previous cycles (and the timetable reference), such a model is called an explicit model. Implicit models can be converted into an explicit model by simply substituting equations into each other as long as no equation of the following form results from the substitution: $x_i(k) = x_i(k) \otimes \tau_i$ where $\tau_i > 0$. In the following example we show how an implicit model can be converted to an explicit model using substitution.

Given the following implicit model

$$x_1(k) = x_1(k-1) \otimes 3 \oplus x_2(k-1) \otimes 3 \tag{48}$$
$$x_2(k) = x_1(k) \otimes 3 \oplus x_1(k-1) \otimes 3 \oplus x_2(k-1) \otimes 3 \tag{49}$$

Equation (48) is already in explicit form, but (49) has an implicit and an explicit part. By substituting (48) into (49) the equation is converted to its explicit form and the result is an explicit model:

$$x_1(k) = x_1(k-1) \otimes 3 \oplus x_2(k-1) \otimes 3 \tag{50}$$
$$\begin{aligned} x_2(k) &= x_1(k-1) \otimes (6 \oplus 3) \oplus x_2(k-1) \otimes (3 \oplus 6) \\ &= x_1(k-1) \otimes 6 \oplus x_2(k-1) \otimes 6 \end{aligned} \tag{51}$$

For an example with only a few, or in this case a single implicit equation this is easy to do. But if there are multiple implicit equations, which will be the case for larger models, this is far from trivial and in some cases an implicit model cannot be converted into an explicit model. In max-plus algebra there is a specific method for the conversion of an implicit into an explicit model by using max-plus matrix multiplication (Theorem 3.17 of Baccelli et al (1992)) . It also gives clear requirements on the implicit model that need to be met to be able to convert it to an explicit model. This method is explained below.

The model of (34) can be rewritten in its explicit form. By doing so the dependency of state vector $x(k)$ on itself is removed.

In general an implicit max-plus-linear model as described by

$$x(k) = r(k) \oplus A_0 \otimes x(k) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} A_\mu \otimes x(k-\mu) \tag{52}$$

can be converted to an explicit max-plus-linear model described by

$$x(k) = A_0^* \otimes \left(r(k) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} A_\mu \otimes x(k-\mu) \right), \tag{53}$$

where A_0^* is defined as ((Baccelli et al 1992)):

$$A_0^* = \bigoplus_{p=0}^{\infty} A_0^{\otimes^p}, \tag{54}$$

with

$$A^{\otimes^p} = \underbrace{A \otimes A \otimes \ldots \otimes A}_{p \text{ times}}, \tag{55}$$

and $A^{\otimes^0} = E$.

However, the model (34) is an implicit *switching* max-plus-linear model, where the matrices depend on the control variables and the values of the different process times (i.e. the dwell times, headway times, separation times, running times, and connection times); furthermore, the matrices $A_\mu(k-\mu)$ with $\mu < 0$ need to be handled. Hence, we cannot directly apply (53)-(55). In the next part of this section we will first describe how the implicit SMPL for a single cycle can be converted into its explicit form and after that the method is expanded to an implicit SMPL for multiple cycles.

4.1 Explicit Model for a Single Cycle

For a model of a single cycle $x(k)$ only needs to be determined and no future cycles, i.e. $x(k-\mu)$ with $\mu < 0$, are considered. As a result, A_μ is only needed for $\mu = 0, \ldots, \mu_{\max}$. This reduces the implicit model to

$$x(k) = r(k) \oplus A_0 \otimes x(k) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} A_\mu \otimes x(k-\mu).$$

Next, A_0^* will need to be determined. Calculating the max-plus matrix powers $A_0^{\otimes^p}$ can be done in the same way as for a constant matrix; the problem is the infinite sum in (54). To tackle this issue we use Theorem 3.20 of Baccelli et al (1992):

Theorem. *(Theorem 3.20 of Baccelli et al (1992))*
If the precedence graph $\mathcal{G}(A)$ has no circuits of positive weight, then

$$A^* = E \oplus A \oplus A^{\otimes^2} \oplus \ldots \oplus A^{\otimes^{n-1}},$$

where n is the dimension of A.

The precedence graph $\mathcal{G}(A)$ is defined as in Definition 2.8 of Baccelli et al (1992):

Definition. *The precedence graph of a square $n \times n$ matrix A with entries in $\mathcal{C}$ is a weighted digraph with n nodes and an arc (j, i) if $A_{i,j} \neq \varepsilon$, in which case the weight of this arc receives the numerical value of $A_{i,j}$. The precedence graph is denoted as $\mathcal{G}(A)$.*

This means that if $A_0^{\otimes^p}$ for $p = 1, \ldots, n$ does not contain any positive diagonal elements then the infinite sum in (54) can be limited to the range $p = 1, \ldots, n-1$.

Note that a positive diagonal element $[A_0^{\otimes^p}]_{ii}$ for $p \in \{1, \ldots, \infty\}$, $i \in \{1, \ldots, n\}$ would describe the following relation:

$$x_i(k) \geq x_i(k) \otimes [A_0^{\otimes^p}]_{ii}.$$

The solution to this equation would then be $x_i(k) = \infty$.

If one or more event times are infinite, the timetable is infeasible. Since the nominal model results in a feasible timetable, infinite event times can only result from changing the control variables. Infinite event times can only have two causes: infinite process times or positive diagonal elements in one of the max-plus matrix powers $A_0^{\otimes^p}$. Since none of the process times in the model can be infinite, the only cause that remains is positive diagonal elements. It is therefore clear that the combinations of control variables resulting in infinite event times also result in positive diagonal elements of $A_0^{\otimes^p}]_{ii}$ for $p \in \{1, \ldots, \infty\}$. Hence, it is necessary to determine the combinations of control variables that result in positive diagonal elements when calculating A_0^* and remove all elements containing these specific combinations of control variables from the matrix powers of $A_0(k, u(k))$. This can be done by simply calculating the matrix powers of A_0 and each time a non-ε diagonal element is in the matrix, check which combination(s) of control variables the diagonal elements consist of and set all elements that have those combinations of control variables to ε. This process is shown in Appendix A.2 for a small example.

By removing these infeasible combinations of inputs from the model a feasible explicit switching max-plus-linear model of the following form is found:

$$x(k) = A_0^{*,\text{feas}} \otimes \left(r(k) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} A_\mu \otimes x(k-\mu) \right), \tag{56}$$

where $A_0^{*,\text{feas}}$ is the feasible part of A_0^*; the infeasible combinations of control variables are removed from the matrix. In Appendix A.2 it is explained how this can be done using a small example.

Some dispatching actions may have an effect on the event times in the next cycle, or even the cycles after that, therefore it may be needed to extend the model to multiple cycles. This will be done in the next section.

4.2 Explicit Model for Multiple Cycles

A model with multiple cycles is used to predict the arrival and departure times for the current and future cycles and the effects of the dispatching actions on these arrival and departure times based on the current situation. That means $x(k)$ and

the event times of future cycles ($x(k-\mu)$, with $\mu < 0$) need to be determined. The model for $m+1$ cycles can be described by the following set of equations:

$$x(k+q) = r(k+q) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} A_\mu \otimes x(k+q-\mu) \oplus \bigoplus_{\mu=-\min(\mu_{\max}, m-q)}^{0} A_\mu \otimes x(k+q-\mu),$$

for the range $q = 0, \ldots, m$. By extending the state vector to

$$\breve{x}(k) = \left[x^\top(k)\ x^\top(k+1)\ \ldots\ x^\top(k+m-1)\ x^\top(k+m)\right]^\top,$$

the above set of equations can be written as

$$\breve{x}(k) = \breve{r}(k) \oplus \breve{A}_0 \otimes \breve{x}(k) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} \breve{A}_\mu \otimes x(k-\mu),$$

where $\breve{A}_0(k, u(k))$ contains the matrices with the process times for the constraints between event times of the current and future cycles ($x(k-\mu)$, with $\mu \leq 0$), and the max-plus sum with μ ranging from 1 to $\mu_{\max}$ contains the matrices with the process times for the constraints between event times of past cycles ($x(k-\mu)$, with $\mu > 0$) and event times of the current and future cycles. These matrices are defined as follows:

$$[\breve{A}_0(k, u(k))]_{i,j} = \begin{cases} A_{i-j}(k+i-1, u(k+j-1)) & \text{if } 0 \leq i-j \leq \mu_{\max} \\ & \text{and } 1 \leq i, j \leq m+1 \\ A_{i-j}(k+i-1, u(k+i-1)) & \text{if } -\mu_{\max} \leq i-j \leq -1 \\ & \text{and } 1 \leq i, j \leq m+1 \\ \mathcal{E} & \text{else,} \end{cases}$$

and

$$\breve{A}_\mu = \left[A_\mu^\top\ A_{\mu+1}^\top\ \ldots\ A_{\mu_{\max}}^\top\ \mathcal{E}^\top\ \ldots\ \mathcal{E}^\top\right]^\top.$$

The reference vector $\breve{r}(k)$ containing the planned event times is defined as follows:

$$\breve{r}(k) = \left[r^\top(k)\ r^\top(k+1)\ \ldots\ r^\top(k+m-1)\ r^\top(k+m)\right]^\top.$$

The same procedure as for the model of one cycle can now be applied to $\breve{A}_0$, resulting in $\breve{A}_0^{*,\text{feas}}$ and a feasible explicit switching max-plus-linear model for multiple cycles:

$$\breve{x}(k) = \breve{A}_0^{*,\text{feas}} \otimes \left(\breve{r}(k) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} \breve{A}_\mu \otimes x(k-\mu)\right) \tag{57}$$

4.3 Structured Approach to Matrix Multiplication

The calculation of the matrix powers of A_0 can be simplified by making use of the structure of the matrix as shown in (21) and the submatrices that are described in Sections 2.5 and 3. First denote $A_{0,4,\mathrm{d}} \oplus A_{0,6,\mathrm{d}}$ as A_a, $A_{0,2} \oplus A_{0,3} \oplus A_{0,5}$ as A_b, $A_{0,1}$ as A_c, and $A_{0,4,\mathrm{a}} \oplus A_{0,6,\mathrm{a}}$ as A_d; then any $A_0^{\otimes m}$ can be split up into four submatrices

$$A_0^{\otimes m} = \begin{bmatrix} A_{1,1,m} & A_{1,2,m} \\ A_{2,1,m} & A_{2,2,m} \end{bmatrix}, \tag{58}$$

with[3]

$$A_{1,1,m} = \bigoplus_{(q_1,q_2,q_3)\in\mathcal{S}_{1,1,m}} \bigotimes_{i=1}^{l} A_\mathrm{a}^{\otimes q_{1,i}} \otimes A_\mathrm{b}^{\otimes q_{2,i}} \otimes A_\mathrm{d}^{\otimes q_{3,i}} \otimes A_\mathrm{c}^{\otimes q_{2,i}}, \tag{59}$$

where $q_{1,i} = (q_1)_i$, $q_{2,i} = (q_2)_i$, $q_{3,i} = (q_3)_i$, and where the set $\mathcal{S}_{1,1,m}$ contains all tuples (q_1, q_2, q_3) of vectors that satisfy the following equation:

$$\sum_{i=1}^{l} q_{1,i} + 2q_{2,i} + q_{3,i} = m,$$

with $q_{1,i} \in \{0, \ldots, m\}$, $q_{2,i} \in \{0, 1\}$, and $q_{3,i} = 0$ if $q_{2,i} = 0$ and $q_{3,i} \in \{0, \ldots, m\}$ if $q_{2,i} = 1$.

For the other submatrices similar equations can be determined:

$$A_{1,2,m} = \bigoplus_{(q_1,q_2,q_3,q_4)\in\mathcal{S}_{1,2,m}} \bigotimes_{i=1}^{l} A_\mathrm{a}^{\otimes q_{1,i}} \otimes A_\mathrm{b}^{\otimes q_{2,i}} \otimes A_\mathrm{d}^{\otimes q_{3,i}} \otimes A_\mathrm{c}^{\otimes q_{4,i}}, \tag{60}$$

where $q_{1,i} = (q_1)_i$, $q_{2,i} = (q_2)_i$, $q_{3,i} = (q_3)_i$, and where the set $\mathcal{S}_{1,2,m}$ contains all tuples (q_1, q_2, q_3, q_4) that satisfy the following equation:

$$\sum_{i=1}^{l} q_{1,i} + 2q_{2,i} - 1 + q_{3,i} = m,$$

where $q_{1,i} \in \{0, \ldots, m\}$, $q_{2,i} \in \{0, 1\}$, $q_{2,l} = 1$, $q_{3,i} = 0$ if $q_{2,i} = 0$ and $q_{3,i} \in \{0, \ldots, m\}$ if $q_{2,i} = 1$, $q_{4,i} = q_{2,i}$, for $i = 1, \ldots, l-1$, and $q_{4,l} = 0$.

$$A_{2,1,m} = \bigoplus_{(q_1,q_2,q_3,q_4)\in\mathcal{S}_{2,1,m}} \bigotimes_{i=1}^{l} A_\mathrm{d}^{\otimes q_{1,i}} \otimes A_\mathrm{c}^{\otimes q_{2,i}} \otimes A_\mathrm{a}^{\otimes q_{3,i}} \otimes A_\mathrm{b}^{\otimes q_{4,i}}, \tag{61}$$

where $q_{1,i} = (q_1)_i$, $q_{2,i} = (q_2)_i$, $q_{3,i} = (q_3)_i$, and where the set $\mathcal{S}_{2,1,m}$ contains all tuples (q_1, q_2, q_3, q_4) that satisfy the following equation:

$$\sum_{i=1}^{l} q_{1,i} + 2q_{2,i} - 1 + q_{3,i} = m,$$

[3] In this equation the exponents of the factors of the matrix product are determined using graph theory: according to graph theory $[A^{\otimes m}]_{i,j}$ corresponds to the maximum weight of all paths of length m from node j to node i in the precedence graph of A. See Baccelli et al (1992) for a detailed review of graph theory.

where $q_{1,i} \in \{0, \ldots, m\}$, $q_{2,i} \in \{0, 1\}$, $q_{2,l} = 1$, $q_{3,i} = 0$ if $q_{2,i} = 0$ and $q_{3,i} \in \{0, \ldots, m\}$ if $q_{2,i} = 1$, $q_{4,i} = q_{2,i}$, for $i = 1, \ldots, l-1$, and $q_{4,l} = 0$.

$$A_{2,2,m} = \bigoplus_{(q_1,q_2,q_3)\in\mathcal{S}_{2,2,m}} \bigotimes_{i=1}^{l} A_{\mathrm{d}}^{\otimes^{q_{1,i}}} \otimes A_{\mathrm{c}}^{\otimes^{q_{2,i}}} \otimes A_{\mathrm{a}}^{\otimes^{q_{3,i}}} \otimes A_{\mathrm{b}}^{\otimes^{q_{2,i}}}, \tag{62}$$

where $q_{1,i} = (q_1)_i$, $q_{2,i} = (q_2)_i$, $q_{3,i} = (q_3)_i$, and where the set $\mathcal{S}_{2,2,m}$ contains all tuples (q_1, q_2, q_3) that satisfy the following equation:

$$\sum_{i}^{l} q_{1,i} + 2q_{2,i} + q_{3,i} = m,$$

where $q_{1,i} \in \{0, \ldots, m\}$, $q_{2,i} \in \{0, 1\}$, $q_{3,i} = 0$ if $q_{2,i} = 0$ and $q_{3,i} \in \{0, \ldots, m\}$ if $q_{2,i} = 1$.

The general idea of how these equations have been determined can be found in Appendix A.4.

The same approach can be applied to $\breve{A}_0$ by reordering the state vector as follows: the event times should be split up in departure and arrival times, the first half of the state vector should consist of the departure times, the second half should consist of the arrival times, and both arrival and departure times should be sorted per track. This results in the same structure for matrix $\breve{A}_0$ as A_0; the dimensions of the submatrices are just larger.

5 Reduction Method

At stations where trains can be reordered, the model can model the change in order of any two trains running over the next track, even if there is a very large time difference between the scheduled departure times. If the maximum of the delays of all trains is known it is possible to determine which (combinations of) control variables will not be used when determining the optimal dispatching actions for reducing the delays. In this section it is explained how these (combinations of) control variables can be determined and removed from the model. First, the delay model will be explained. After that the reduction method is explained using the delay model.

5.1 Delay Model

The model as defined in Section 3 has a state vector $x(k)$ that corresponds to the arrival and departure times of the trains. When dealing with delays and trying to minimize them it can be more useful to transform the model such that the transformed state vector $x^{\mathrm{d}}(k)$ shows the delays instead of the arrival and departure times of the trains. Another advantage of this model transformation is that the elements of the transformed matrix A_{μ}^{d} are the negative slack times between the events. The concepts of the delay model and negative slack time are based on slack time, realizability and structural delays as described by Goverde (2010) and Hansen and Pachl (2008). The slack time is defined in Hansen and Pachl (2008) as:

Definition 1. *For any activity (j,i) the slack time is the difference of the end $x_j(k-\mu_{ij})+a_{ij}(k,u(k-\mu_{ij})$ of the activity and the start $x_i(k)$ of the new activity.*

where an activity (j,i) is described by one of the constraints introduced in Section 3. The slack time can be used to analyze the model on robustness against delays, and in the perturbed mode, it can be used to analyze the effects of the different dispatching actions.

The matrix A_μ is transformed into A_μ^{d}, containing the negative slack times:

$$[A_\mu^{\mathrm{d}}]_{i,j} = [A_\mu]_{i,j} - (r_i(0) - (r_j(0) - \mu T)) \tag{63}$$

where we used

$$r(k) = r(0) + kT. \tag{64}$$

With these matrices the model from (34) can be transformed into

$$x^{\mathrm{d}}(k) = \mathbf{0} \oplus \bigoplus_{\mu=0}^{\mu_{\max}} A_\mu^{\mathrm{d}}(k,u(k-\mu)) \otimes x^{\mathrm{d}}(k-\mu) \oplus \bigoplus_{\mu=-\mu_{\max}}^{-1} A_\mu^{\mathrm{d}}(k,u(k)) \otimes x^{\mathrm{d}}(k-\mu), \tag{65}$$

where the state vector $x^{\mathrm{d}}(k)$ contains the delays of the events of cycle k and $\mathbf{0}$ is a vector of the same length as $r(k)$ filled with zeros.

Because the elements of the matrix A_μ^{d} represent the negative slack times between events, the value of the elements shows how much one event is delayed directly by another event. For example, if $[A_0^{\mathrm{d}}]_{i,j} = 2$, then $x_i(k)$ is delayed by at least 2 minutes by $x_j(k)$. Therefore, during nominal operation all the elements of the matrices should be negative, otherwise there would be delays during nominal operation.

5.2 Removing Redundant Control Variables

In the previous subsection the model was rewritten into a delay model, where the entries of the matrix A_μ^{d} contain the negative slack times. The negative slack times can be used as a measure of the effects of the control variables. Theorem 1 of Kersbergen et al (2013b) states:

Theorem. *(Theorem 1 of Kersbergen et al (2013b)) The elements of the matrix powers of A_0^d give lower bounds to the delays caused by the control variables, if and only if, the process times used are the minimal process times.*

The minimal process times are the smallest possible process times that are achievable by the trains and the railway operations.

By using the delay model together with the minimal process times the minimum delays caused by the (combinations of) control variables can be determined. Then by assuming there is a known maximum value for the delays, the (combinations of) control variables, that would result in delays larger than the maximum delay, can be removed. Removing these (combinations of) control variables will have no effect on the solution of the dispatching problem, but it will reduce the complexity of the problem. These value combinations of control variables can be

removed in the same way as the control variables for infeasible train orders are removed, which was shown in Appendix A.2.

These (combinations of) control variables can be found by determining which of the control variables result in at least one element of the max-plus powers of $\breve{A}_0^{\mathrm{d}}$ being larger than the maximum value for the delays. These elements can be determined off-line by making use of Theorem 1 and the minimal process times. By using the minimal process times the values of the max-plus powers of $\breve{A}_0^{\mathrm{d}}$ are a lower bound of the delays caused by the different (combinations of) control variables. The reduction is done by replacing these (combinations of) control variables by ε in all elements of the max-plus powers of $\breve{A}_0^{\mathrm{d}}$, effectively removing them from the model. The reduction can be applied while calculating the feasible explicit model resulting in a reduced explicit model:

$$\breve{x}^{\mathrm{d}}(k) = \breve{A}_0^{\mathrm{d},*,\mathrm{red}} \otimes \left(\breve{\mathbf{0}}(k) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} \breve{A}_\mu^{\mathrm{d},\mathrm{red}} \otimes x^{\mathrm{d}}(k-\mu) \right). \tag{66}$$

This reduction method can also be applied to the implicit model, but it will not be as effective, because most combinations of control variables are only modeled explicitly in the max-plus matrix powers of $\breve{A}_0^{\mathrm{d},*,\mathrm{red}}$ and can therefore not be removed from the implicit model. In fact, in the implicit model most combinations of control variables are modeled implicitly through the dependency of $x(k)$ on itself.

In Kersbergen et al (2013a) it was shown how an SMPL model can be used for the dispatching problem. The dispatching problem is the problem of minimizing a measure of the delays (for example the sum of delays) using the available dispatching actions (reorder trains, switch tracks, break connections and break joined trains) by predicting the effects of the dispatching actions on the future events of the railway network. This problem can be described as a MILP problem, where the SMPL is used for the prediction of the effects of the dispatching actions. The model is converted to suite the MILP format by replacing the max-plus control variables by binary variables and converting all maximizations back to separate constraints. For a more detailed explanation the reader is referred to Kersbergen et al (2013a).

In the next section we will show the effects of the reduction method on the computation time.

6 Case Study

In this section the effects of the reduction method on the solution time of the rescheduling problem will be evaluated. Furthermore, the effects of different objective functions on the found solution and computation time will be evaluated via a case study. In this case study we will look at a single step of the model predictive controller.

The case study will be based on the Dutch Railway network and the timetable of the year 2006, since we have accurate process times available for this timetable. The model is simplified because only intercity and interregional trains are considered. Furthermore, only stations and junctions are considered where the trains can be rescheduled. Arrivals and departures at stations where trains can only stop and not overtake are not explicitly modeled. The only dispatching actions in this

case study are the reordering of trains. The timetable period is one hour and the railway traffic is considered for a single hour.

Delays in the network can be divided into two types: primary and secondary delays. For each train the primary delays are the delays that cannot be avoided. They are caused by increased process times of that train or because of trains that hinder it and cannot be avoided by rescheduling. The secondary delays of a train are all delays that can be avoided by applying dispatching actions. For the entire network none of the primary delays can be recovered with the use of dispatching actions. Only some of the secondary delays can be recovered with dispatching actions. Not all secondary delays can be recovered since some dispatching actions may reduce the secondary delays of one train, but increase the secondary delays of another train. The goal is to minimize the secondary delays.

To test the effectiveness of the rescheduling method and the computation time needed for the implicit and reduced explicit model we have built a set of 500 delay scenarios. In each scenario 20% of the trains in the first hour are randomly selected and given a random primary delay by increasing the running time of those trains. The value of the delay is determined by a Weibull distribution with scale parameter 6 and shape parameter 0.8. The model predictive controller then optimizes the dispatching actions for the next hour. In this hour no new primary delays are introduced. The only delays present in this hour are the delays that propagated from the primary delays in the previous period. The same is also done with a model predictive controller that optimizes the dispatching actions for the next two hours. Determining the optimal dispatching actions can be done by solving an MILP problem.

The value for the maximum delay used in the reduction method is set to 15 minutes.

All calculations are done on an AMD Phenom II X4 960T at 3GHz with 16GB of memory, running Windows 7 64bit. The model is built up in MATLAB and all solvers are called using the mex-interface of MATLAB. The solvers used are GLPK 4.46 (GLPK (2014)), Gurobi 5.60 (Gurobi (2014)) and TOMLAB /CPLEX 12.5 (TOMLAB (2014)).

6.1 Minimization of the Sum of Delays

In general an MILP problem has the following structure:

$$\begin{aligned} &\min_{z} c\,z \\ &\text{s.t. } A\,z \leq b \end{aligned}$$

where z is a set of mixed (continuous, integer and binary) variables that can be adjusted in order to minimize the cost function $c\,z$, while the variables must satisfy the constraints in $A\,z \leq b$. In this case study $z = [x^\top, \nu^\top]^\top$,where x contains the arrival and departure delays of $x(k)$ for the hour in which the controller is active and ν contains the binary variables corresponding to the max-plus binary variables $u(k), v(k)$, and $w(k)$ for this period of time. For the exact details on how to write the dispatching problem as an MILP problem we refer to Kersbergen et al (2013a). The cost-function is $c\,z = [\mathbf{1}\, 0.0001 \times \mathbf{1}]\,[x^\top\, \nu^\top]^\top$, where $\mathbf{1}$ is a row vector of appropriate size containing only ones. It is the sum of all delays and a very small

weight is put on the control variables to ensure that if multiple solutions result in the same sum of delays the solution with the least changes to the schedule is chosen.

If information is known about the number of passengers that get on and off at each station, this information could be used to weigh the delays by changing $\mathbf{1}$ into a vector with different weights for different continuous variables. This would give a measure of passenger delays instead of train delays. Another possible cost function would be the reduction of the maximum secondary delay, but this would require the model to be adapted in a similar way as is done by Corman et al (2012), such that a continuous variable represents this maximum secondary delay. For more complex case studies including multiple kinds of dispatching actions the weights on the control actions could be changed such that breaking connections are weighted against the increase in delays.

The term $A\,z \leq b$ describes the constraints of the implicit and explicit models respectively. Because the exact set of trains and dispatching actions that are considered depends on the delay scenario, the size of the constraint matrices vary. The general structure of the matrices remains the same for the different delay scenarios and is shown in Figures 2 and 3 for the implicit and explicit MILP problem. The constraints in the implicit MILP problem have one or two continuous variables. The constraints in the explicit MILP problem all have one continuous variable, except for the constraints used to ensure that certain combinations of control variables are not chosen, such as the combinations that result in infeasible train order and the combinations of control variables removed by the reduction method. Those constraints have no continuous variables in them.

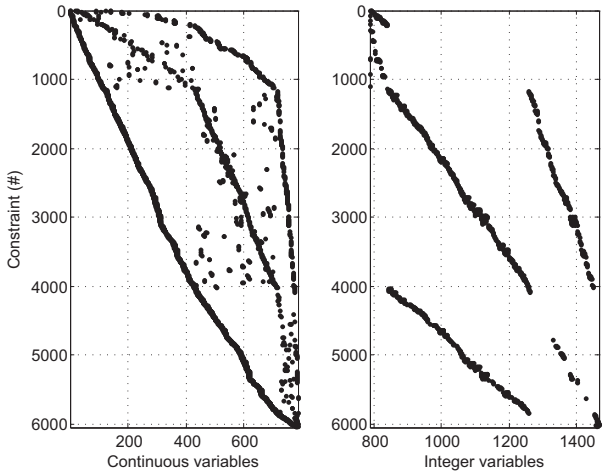

Fig. 2 Structure of the constraint matrix of the implicit MILP problem.

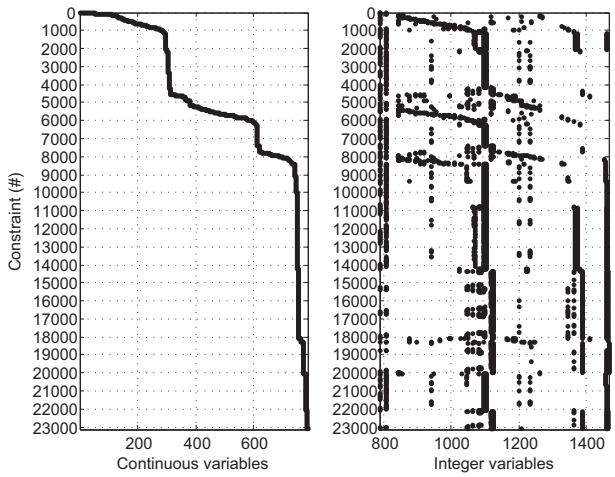

Fig. 3 Structure of the constraint matrix of the explicit MILP problem.

6.1.1 Prediction Horizon of One Hour

The average, minimum, and maximum number of continuous variables, binary variables, and constraints for the implicit and explicit model for a prediction horizon of one hour are given in Table 1.

Table 1 Number of constraints, continuous and binary variables of the MILP problems

	Average	min	max
Continuous variables	773.74	759	790
Binary variables	670.74	633	711
Constraints-Implicit	5946	5794	6118
Constraints-Explicit	3591	3109	4325

For the 500 scenarios we will first look at how much the secondary delays are reduced by applying control. This is shown in Figure 4. On average the reduction in secondary delays is 34.17%. In three scenarios there was no reduction in delays.

For the distribution of the delays we only consider the events that have a non-zero delay in the uncontrolled and/or the controlled case. For these events the distributions are shown in Figure 5 for the uncontrolled and controlled case. By comparing the two distributions, it is clear that in the controlled case several events are no longer delayed. From the comparison of the distributions in Figure 5, it is also clear that in the controlled case there are more short delays and less longer delays. The longest delays are also a couple of minutes bigger for the controlled case. But there are very few of those delays.

Next we will look at the computation time of the solution of the MILP problems with the use of the solvers GLPK 4.46, CPLEX 12.5, and Gurobi 5.60 for

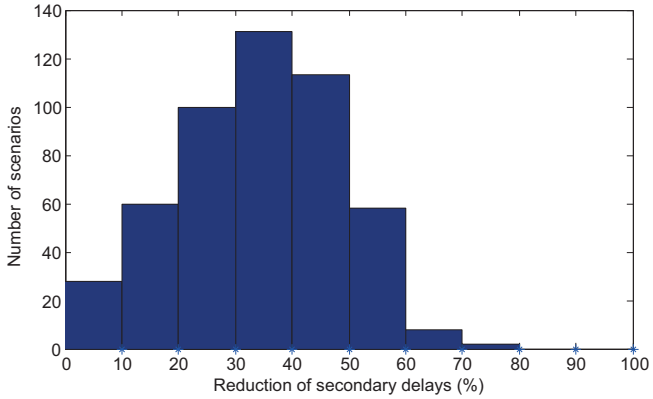

Fig. 4 Histogram of the reduction of secondary delay for the 250 scenarios.

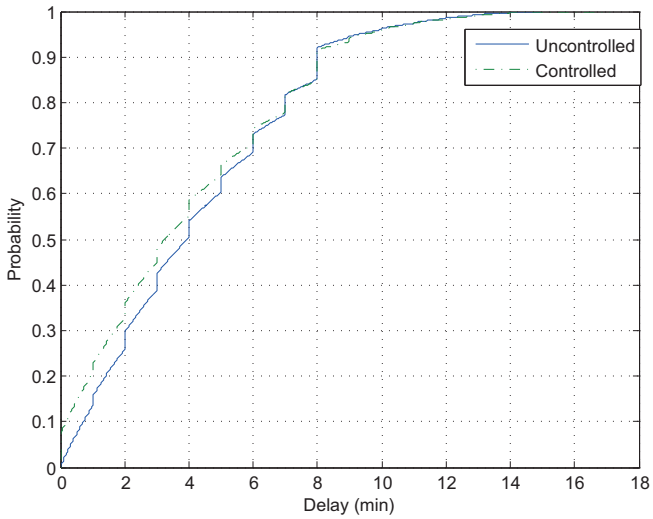

Fig. 5 Cumulative distribution of the delays for the controlled and uncontrolled case.

the implicit and explicit models. Box plots of the computation times for the 500 scenarios for the implicit and explicit MILP problem are given in Figure 6. Statistics on the computations are given in Table 2 for CPLEX, Gurobi, and GLPK. The mex-interface of GLPK did not provide any solver statistics except for the computation time. The statistics that are given are the number of simplex iterations and the computation time. The integrality gap is also given, which is not a solver statistic, but a statistic of the MILP problem. It is the objective value of the optimal solution of the MILP problem divided by the objective value of the optimal solution of the linear programming relaxation of the MILP problem. The minimum value of the integrality gap is one, since the objective value of the optimal solution of the MILP problem can never be lower than the objective value of the optimal solution of the linear programming relaxation of the MILP problem.

In general it is assumed that if the integrality gap is closer to one the problem is easier to solve.

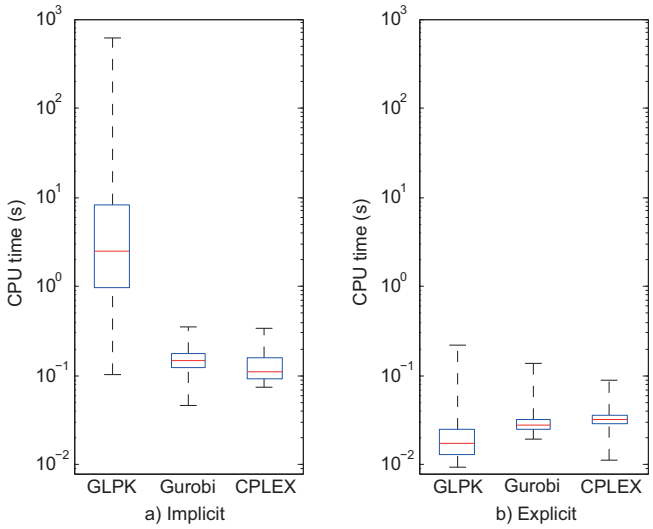

Fig. 6 Computation time of the MILP solvers for the implicit and explicit MILP problem for a one hour prediction horizon.

Table 2 Computation statistics for the given MILP solvers

	Implicit			Explicit		
	avg	min	max	avg	min	max
Comp. time (s) (GLPK)	11.93	0.1014	603.92	0.0233	0.0096	0.2161
Comp. time (s) (Gurobi)	0.1477	0.0459	0.3494	0.0315	0.0196	0.1358
Comp. time (s) (CPLEX)	0.1266	0.0754	0.3400	0.0331	0.0112	0.0902
Simplex iterations (Gurobi)	159.72	17	869	49.06	0	411
Simplex iterations (CPLEX)	119.7	14	617	27.97	0	232
Integrality Gap	1.0813	1.0148	1.2158	1.0715	1.0135	1.1757

For the GLPK solver the difference in computation time of the explicit MILP problem compared to the explicit MILP problem is the largest. The explicit MILP problem is solved 513 times faster on average than the implicit model. For the Gurobi solver the difference is much smaller. The explicit MILP problem is solved only 4.69 times faster on average. On average the Gurobi solver needs to solve 3.26 times less simplex iterations. For the CPLEX solver we see a similar picture. The computation time is on average about 3.83 times faster. The number of simplex iterations that need to be solved is on average 4.28 times higher for the implicit MILP problem. The distance of the integrality gap to the value one is 12.1% lower for the explicit model compared to the implicit model (0.0715 compared to 0.0813). When we compare the fastest implicit solver (CPLEX) with the fastest explicit

solver (GLPK), then the computation time needed to solve the implicit MILP problem is 5.44 times higher.

6.1.2 Prediction Horizon of Two Hours

For the implicit and explicit MILP problems based on the prediction horizon of two hours the average, minimum, and maximum number of continuous variables, binary variables, and constraints for are given in Table 3.

Table 3 Number of constraints, continuous and binary variables of the MILP problems

	Average	min	max
Continuous	1542.9	1530	1559
Binary	2347.8	2307	2400
Constraints-Implicit	14037	13870	14233
Constraints-Explicit	50981	46044	57332

For the prediction horizon of two hours we will only look at the computation time and solver statistics, since the reduction of delays and the distribution of the delays are similar to those shown for the prediction horizon of one hour. The computation times for the 500 scenarios for the implicit and explicit MILP problem are given as box plots in Figure 7.

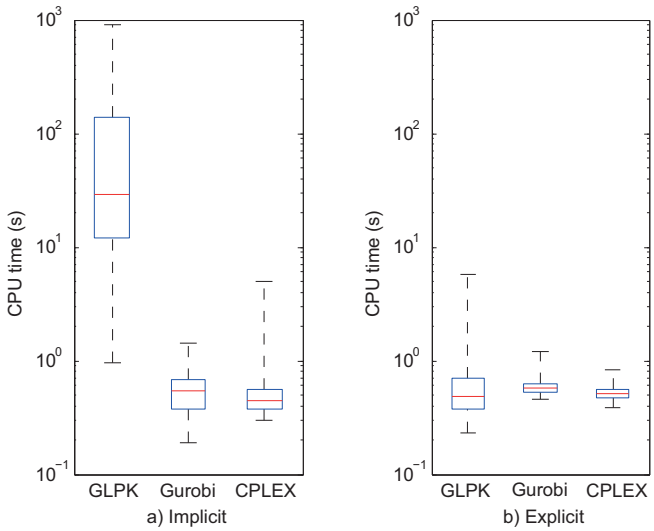

Fig. 7 Computation time of the MILP solvers for the implicit and explicit MILP problem for a two hour prediction horizon.

Statistics on the computation time for CPLEX, Gurobi, and GLPK are given in Table 4.

Table 4 Computation statistics for the given MILP solvers

	Implicit			Explicit		
	avg	min	max	avg	min	max
Comp. time (s) (GLPK)	155.8	0.9599	903.7	0.6132	0.2286	5.792
Comp. time (s) (Gurobi)	0.5451	0.1870	1.4355	0.5943	0.4515	1.1903
Comp. time (s) (CPLEX)	0.5019	0.3007	5.0470	0.5247	0.3900	0.8200
Simplex iterations (Gurobi)	188.9	39	1175	525.2	0	1633
Simplex iterations (CPLEX)	242.2	55	923	71.98	0	262
Integrality Gap	1.119	1.022	1.267	1.100	1.021	1.228

In this case the average computation time of the explicit model for GLPK is 254 times lower, while the maximum is 156 times lower. For Gurobi the explicit model is solved 1.09 times slower on average. The number of simplex iterations is much higher for the explicit model. The increased computation time and number of simplex iterations is due to the increased size of the problem. The number of constraints is, on average, 3.63 times higher for the explicit model. For CPLEX the number of simplex iterations is lower for the explicit model, but the average computation time is still 1.04 times higher. This is again due to the increased size of the constraint matrix and as a result the simplex iterations take more time to be completed. The maximum computation time however is 6.15 times lower for CPLEX when solving the explicit MILP problem. The distance of the integrality gap to the value one is 16.0% lower for the explicit model compared to the implicit model (0.100 compared to 0.119)

6.2 Minimization of the Sum of Arrival Delays

When considering the delay in the network it can make more sense to only consider one delay per train at each station, so only the arrival or the departure delay at the station. Since passengers are mostly interested in the time they arrive we will consider minimizing the sum of arrival delays as the cost function.

If we look at the structure of the explicit switching max-plus-linear model as described in (57) and repeated here for convenience:

$$\breve{x}(k) = \breve{A}_0^{*,\mathrm{feas}} \otimes \left(\breve{r}(k) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} \breve{A}_\mu \otimes x(k-\mu) \right)$$

And we expand it by splitting the state vector up into the arrival and departure delays:

$$\breve{x}(k) = \breve{A}_0^{*,\mathrm{feas}} \otimes \breve{r}(k) \oplus \bigoplus_{\mu=1}^{\mu_{\max}} \breve{A}_{\mathrm{exp},\mu} \otimes x(k-\mu)$$

$$\begin{bmatrix} \breve{d}(k) \\ \breve{a}(k) \end{bmatrix} = \begin{bmatrix} \breve{A}_{0,\mathrm{a}}^{*,\mathrm{feas}} & \breve{A}_{0,\mathrm{b}}^{*,\mathrm{feas}} \\ \breve{A}_{0,\mathrm{c}}^{*,\mathrm{feas}} & \breve{A}_{0,\mathrm{d}}^{*,\mathrm{feas}} \end{bmatrix} \otimes \begin{bmatrix} \breve{r}_{\mathrm{d}}(k) \\ \breve{r}_{\mathrm{a}}(k) \end{bmatrix} \oplus \bigoplus_{\mu=1}^{\mu_{\max}} \begin{bmatrix} \breve{A}_{\mathrm{exp},\mu,\mathrm{a}} & \breve{A}_{\mathrm{exp},\mu,\mathrm{b}} \\ \breve{A}_{\mathrm{exp},\mu,\mathrm{c}} & \breve{A}_{\mathrm{exp},\mu,\mathrm{d}} \end{bmatrix} \begin{bmatrix} \breve{d}(k-\mu) \\ \breve{a}(k-\mu) \end{bmatrix}$$

where $\breve{A}_{\text{exp},\mu} = \breve{A}_0^{*,\text{feas}} \otimes \breve{A}_\mu$, and

$$\breve{A}_0^{*,\text{feas}} = \begin{bmatrix} \breve{A}_{0,\text{a}}^{*,\text{feas}} & \breve{A}_{0,\text{b}}^{*,\text{feas}} \\ \breve{A}_{0,\text{c}}^{*,\text{feas}} & \breve{A}_{0,\text{d}}^{*,\text{feas}} \end{bmatrix}$$

$$\breve{A}_{\text{exp},\mu} = \begin{bmatrix} \breve{A}_{\text{exp},\mu,\text{a}} & \breve{A}_{\text{exp},\mu,\text{b}} \\ \breve{A}_{\text{exp},\mu,\text{c}} & \breve{A}_{\text{exp},\mu,\text{d}} \end{bmatrix}$$

This can be split into two equations, one to calculate the arrivals and one to calculate the departures:

$$\breve{d}(k) = \begin{bmatrix} \breve{A}_{0,\text{a}}^{*,\text{feas}} & \breve{A}_{0,\text{b}}^{*,\text{feas}} \end{bmatrix} \otimes \begin{bmatrix} \breve{r}_\text{d}(k) \\ \breve{r}_\text{a}(k) \end{bmatrix} \oplus \bigoplus_{\mu=1}^{\mu_\text{max}} \begin{bmatrix} \breve{A}_{\text{exp},\mu,\text{a}} & \breve{A}_{\text{exp},\mu,\text{b}} \end{bmatrix} \begin{bmatrix} \breve{d}(k-\mu) \\ \breve{a}(k-\mu) \end{bmatrix} \tag{67}$$

$$\breve{a}(k) = \begin{bmatrix} \breve{A}_{0,\text{c}}^{*,\text{feas}} & \breve{A}_{0,\text{d}}^{*,\text{feas}} \end{bmatrix} \otimes \begin{bmatrix} \breve{r}_\text{d}(k) \\ \breve{r}_\text{a}(k) \end{bmatrix} \oplus \bigoplus_{\mu=1}^{\mu_\text{max}} \begin{bmatrix} \breve{A}_{\text{exp},\mu,\text{c}} & \breve{A}_{\text{exp},\mu,\text{d}} \end{bmatrix} \begin{bmatrix} \breve{d}(k-\mu) \\ \breve{a}(k-\mu) \end{bmatrix} \tag{68}$$

In these equations the arrival delays only depend the arrival and departure delays of the previous cycles and not on the departure delays of the current cycle. If we only want to determine the arrival delays, for example in the case we are only interested in minimizing the sum of arrival delays, then we do not need to calculate the departure delays.

In this case the cost function of the MILP problem becomes:

$$c\,z = [\mathbf{0}\,\mathbf{1}\,0.0001 \times \mathbf{1}] \begin{bmatrix} d(k) \\ a(k) \\ \nu(k) \end{bmatrix}.$$

We do not need the departure delays and therefore we can simply calculate the arrival delay. For the explicit MILP problem this means that the constraint matrix A only needs to consists of the constraints from (68), and the constraints from the reduction method and infeasible train orders. This effectively reduces the size of the explicit MILP problem.

We have generated 250 new scenarios, using the same parameters as in the previous case study, and have compared the computation time needed to solve the implicit and explicit MILP problems again for one and two hour prediction horizons. In this case study we will only look at the computation time and computational statistics of the solvers, since the distribution and reduction of the delays are again similar to the distribution and reduction in the first case study.

6.2.1 One hour prediction horizon

The specifications of the MILP problems for a prediction horizon of one hour are in Table 5.

The explicit MILP problem has about half the continuous variables since it only needs to determine the arrival delays thanks to the explicit model structure, the implicit MILP problem has to determine all delays since the arrival delays depend on the arrival and departure delays of other trains through the implicit constraints. Because of the reduction method and the lower number of continuous variables the explicit MILP problem needs to consider the number of constraints

Table 5 Number of constraints, continuous and binary variables of the MILP problems.

	Implicit			Explicit		
	avg	min	max	avg	min	max
Continuous variables	773.96	759	792	385.48	378	393
Binary variables	671.36	639	710	671.36	639	710
Constraints	5950	5794	6252 6162	2210	1954	3061

is on average 2.69 times lower than the number of constraints of the implicit MILP problem. We therefore expect the explicit MILP problem to be solved faster than the implicit MILP problem. The computation time needed to solve the implicit and explicit MILP problem for the three solvers is shown in Figure 8. The computation time, number of simplex iterations, and the integrality gap for the different MILP problems for CPLEX, Gurobi, and GLPK are given in Table 6.

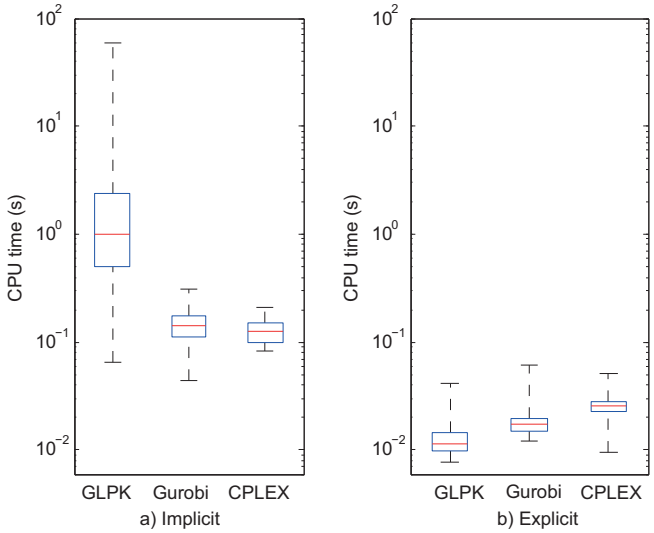

Fig. 8 Computation time of the MILP solvers for the implicit and explicit MILP problem for the sum of arrival delays for a prediction horizon of one hour.

Table 6 Computation statistics for the given MILP solvers.

	Implicit			Explicit		
	avg	min	max	avg	min	max
Comp. time (s) (GLPK)	3.4993	0.0648	59.03	0.0128	0.0413	0.0078
Comp. time (s) (Gurobi)	0.1438	0.0438	0.3109	0.0122	0.0173	0.0610
Comp. time (s) (CPLEX)	0.1284	0.0836	0.2090	0.0259	0.0095	0.0506
Simplex iterations (Gurobi)	168.18	17	524	20.84	0	162
Simplex iterations (CPLEX)	103.39	14	331	15.27	0	66
Integrality Gap	1.098	1.026	1.181	1.086	1.023	1.169

From these results we can conclude that the difference in computation time between the explicit and implicit MILP problem is the largest for GLPK. The explicit MILP problem is solved 272.7 times faster on average. For Gurobi the difference is much smaller, but still significant, with the implicit MILP problem being solved 7.8 times slower on average. The number of simplex relaxations that need to be solved is on average 8.1 times higher for the implicit model. With CPLEX the implicit MILP problem is solved 5.0 times slower than the explicit MILP problem on average. The difference in the number of simplex relaxations that need to be solved is on average 6.8 times lower for the explicit model. When we compare the fastest implicit MILP solver (CPLEX) with the fastest explicit MILP solver (GLPK) the solution is found 10.0 times faster on average using the explicit MILP problem with GLPK. The distance of the integrality gap to the value one is 11.8% lower for the explicit model compared to the implicit model (0.086 compared to 0.098)

6.2.2 Two hour prediction horizon

For the two hour prediction horizon the number of constraints, continuous and binary variables of the MILP problems are given in Table 7.

Table 7 Number of constraints, continuous, and binary variables of the MILP problems.

	Implicit			Explicit		
	avg	min	max	avg	min	max
Continuous variables	1542.8	1530	1559	769.14	762	776
Binary variables	2347.1	2307	2382	2347.1	2307	2382
Constraints	14035	13870	14212	28791	26208	32092

Due to the increased size of the prediction horizon the number of constraints, continuous, and binary variables have increased. The number of constraints of the explicit MILP problems is now 2.05 times higher than the number of constraints in the implicit model. This will affect the computation time the solvers of the explicit MILP problems. The computation time needed to solve the implicit and explicit MILP problem for the three solvers is given in Figure 9. The computation time, number of iterations and number of nodes explored for the different MILP problems for CPLEX and Gurobi are given in Table 8.

Table 8 Computation statistics for the given MILP solvers

	Implicit			Explicit		
	avg	min	max	avg	min	max
Comp. time (s) (GLPK)	65.58	0.3904	902.7	0.2043	0.1484	0.4157
Comp. time (s) (Gurobi)	0.5906	0.2041	1.2246	0.3552	0.2862	0.8465
Comp. time (s) (CPLEX)	0.5213	0.3011	5.1109	0.2656	0.1328	0.4099
Simplex iterations (Gurobi)	227.0	31	1069	275	0	608
Simplex iterations (CPLEX)	224.5	46	642	33.42	0	128
Integrality Gap	1.137	1.001	1.320	1.115	1.000	1.234

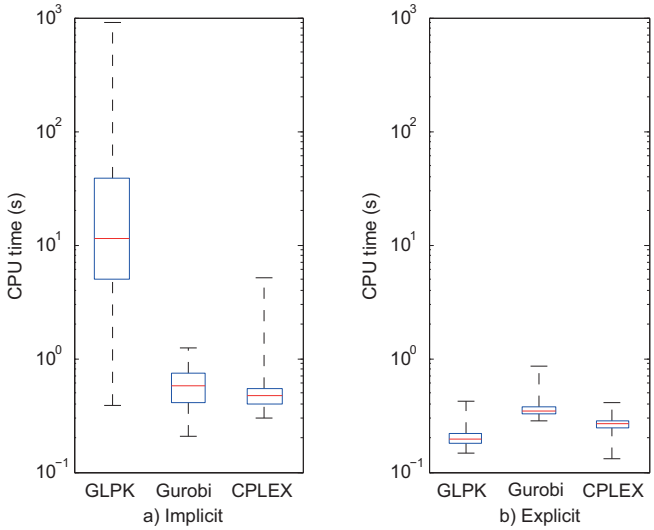

Fig. 9 Computation time of the MILP solvers for the implicit and explicit MILP problem for the sum of arrival delays for a prediction horizon of two hours.

The GLPK solver solves the explicit MILP problems the fastest. It solves the explicit MILP problems 321 times faster on average than the implicit MILP problems. Gurobi solves the explicit MILP problems 1.66 times faster than the implicit MILP problems on average. CPLEX solves the explicit MILP problems 1.96 times faster than the implicit MILP problems on average. The fastest solver for the implicit MILP problems is CPLEX with an average computation time of 0.5213 seconds. The fastest solver for the explicit MILP problems is GLPK with an average computation time of 0.2043 seconds. This is 2.55 times faster. This may be explained by the lower integrality gap of the explicit model. The distance of the integrality gap to the value one is 16.1% lower for the explicit model compared to the implicit model (0.115 compared to 0.137).

7 Conclusions

In this paper it has been shown how to model railway traffic and dispatching actions such as reordering trains, changing tracks, breaking connections and breaking or joining trains, using the max-plus algebra.

With the use of a theory from max-plus algebra, which is only valid for max-plus models, the implicit switching max-plus linear model can be easily converted into its explicit form. The explicit model can be effectively reduced in size, resulting in a decrease in the computation time needed to solve the rescheduling problem using this model compared to the implicit model. For a model consisting of the largest part of the Dutch railway network the average computation time of the rescheduling problem was reduced by a factor 5.44 for a one hour prediction horizon when trying to minimize the sum of delays and by a factor 10 when trying to minimize the sum of arrival delays. For a two hour prediction horizon the

average computation time did not decrease for the sum of delays, but the maximum computation time was 6.15 times lower. For the sum of arrival delays the computation was reduced by a factor 2.55. The proposed conversion to the explicit model and subsequent reduction of the model clearly reduces the time needed to solve the dispatching problem using that model.

The next step in our research will be to consider even larger dispatching problems, such as the whole Dutch railway network for a period of several hours. Another step will be to solve the dispatching problem using distributed model predictive control, in order to further reduce the computation time. By splitting up the model into several smaller models we can take advantage of the explicit model structure to further improve the computation time. Some solvers work faster when the constraint matrix has a specific structure, such a block angular structure. We expect that if we reorder the constraint matrix some of the solvers will be able to solve the problem faster. In our future work we will therefore investigate the effects of reordering the constraint matrix of the MILP according to the track layout or per train line. More research should also be done on the other subsystems of the dispatching support system such as the monitoring and the route and trajectory optimization.

Acknowledgements This research is supported by the Dutch Technology Foundation STW, project 11025 "Model-Predictive Railway Traffic Management; A Framework for Closed-Loop Control of Large-Scale Railway Systems". STW is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs. This research is also funded by TÁMOP-4.2.1./B-11/2/KMR-2011-002 and TÁMOP-4.2.2./B-10/1-2010-0014.

Appendix

A.1 Headway Constraints with Control Variables

If $u_{il}(k-\mu_{i,l}) = 0$ in (37)-(40), then $\overline{u_{il}(k-\mu_{i,l})} = \varepsilon$ and the equations become:

$$\begin{aligned}
d_i(k) &\geq d_l(k-\mu_{i,l}) \otimes \tau_{\mathrm{h,d},i,l}(k) \otimes 0 = d_l(k) \otimes \tau_{\mathrm{h,d},i,l}(k) \\
a_i(k) &\geq a_l(k-\mu_{i,l}) \otimes \tau_{\mathrm{h,a},i,l}(k) \otimes 0 = a_l(k) \otimes \tau_{\mathrm{h,a},i,l}(k) \\
d_l(k-\mu_{i,l}) &\geq d_i(k) \otimes \tau_{\mathrm{h,d},l,i}(k-\mu_{i,l}) \otimes \varepsilon = \varepsilon \\
a_l(k-\mu_{i,l}) &\geq a_i(k) \otimes \tau_{\mathrm{h,d},l,i}(k-\mu_{i,l}) \otimes \varepsilon = \varepsilon.
\end{aligned}$$

The first two equations are identical to equations (5) and (6), and the last two equations simply state that $d_l(k-\mu_{i,l})$ and $a_l(k-\mu_{i,l})$ should be larger than ε, which they always are. This results in the default order of the train runs: first l, then i.

If $u_{i,l}(k-\mu_{i,l}) = \varepsilon$, then $\overline{u_{i,l}(k-\mu_{i,l})} = 0$ and the equations become:

$$\begin{aligned}
d_i(k) &\geq d_l(k-\mu_{i,l}) \otimes \tau_{\mathrm{h,d},i,l}(k) \otimes \varepsilon = \varepsilon \\
a_i(k) &\geq a_l(k-\mu_{i,l}) \otimes \tau_{\mathrm{h,a},i,l}(k) \otimes \varepsilon = \varepsilon \\
d_l(k-\mu_{i,l}) &\geq d_i(k) \otimes \tau_{\mathrm{h,d},l,i}(k-\mu_{i,l}) \otimes 0 = d_i(k) \otimes \tau_{\mathrm{h,d},l,i}(k-\mu_{i,l}) \\
a_l(k-\mu_{i,l}) &\geq a_i(k) \otimes \tau_{\mathrm{h,d},l,i}(k-\mu_{i,l}) \otimes 0 = a_i(k) \otimes \tau_{\mathrm{h,a},l,i}(k-\mu_{i,l})
\end{aligned}$$

Now the first two equations simply state that $d_i(k)$ and $a_i(k)$ should be larger than ε, which they always are, and the last two equations are equal to equations (35) and (36). This results in the changed order of train runs: first i, then l.

A.2 Infeasible Train Order

Consider the headway constraints of three trains running over the same track in the same cycle. Let us assume that the headway times between all three trains are 3 minutes, then the headway constraints between the departures are given by

$$A_{0,4,\mathrm{d}}(u(k),k) = \begin{bmatrix} \varepsilon & 3 \otimes \overline{u_1(k)} & 3 \otimes \overline{u_3(k)} \\ 3 \otimes u_1(k) & \varepsilon & 3 \otimes \overline{u_2(k)} \\ 3 \otimes u_3(k) & 3 \otimes u_2(k) & \varepsilon \end{bmatrix}$$

where $u_1(k)$ determines the order between train 1 and 2, u_2 determines the order between train 2 and 3, and $u_3(k)$ determines the order between train 1 and 3. Now if we set $u_1(k) = 0$, then train 1 traverses the track before train 2; if we set $u_2(k) = 0$, then train 2 traverses the track before train 3; and if we set $u_3(k) = \varepsilon$, then train 3 traverses the track before train 1. But this is impossible because if train 1 traverses the track before train 2 and 2 traverses the track before 3 then $u_1 = 0$ together with $u_2 = 0$ implies that train 1 traverses the track before train 3, which is exactly the opposite of what $u_3 = \varepsilon$ implies. Clearly the combination $u_1 = 0$, $u_2 = 0$, and $u_3 = \varepsilon$ is an infeasible combination and can be removed from the model without having an effect on the feasible train orders.

These infeasible train orders can be derived from the matrix powers of $A_{0,4,\mathrm{d}}$:

$$\begin{aligned} A^{\otimes^2}_{0,4,\mathrm{d}}(u(k),k) &= \begin{bmatrix} 6 \otimes \overline{u_1(k)} \otimes u_1(k) & 6 \otimes u_2(k) \otimes \overline{u_3(k)} & 6 \otimes \overline{u_1(k)} \otimes \overline{u_2(k)} \\ 6 \otimes \overline{u_2(k)} \otimes u_3(k) & 6 \otimes \underline{u_2(k)} \otimes \overline{u_2(k)} & 6 \otimes u_1(k) \otimes \overline{u_3(k)} \\ 6 \otimes u_1(k) \otimes u_2(k) & 6 \otimes \overline{u_1(k)} \otimes u_3(k) & 6 \otimes u_3(k) \otimes \overline{u_3(k)} \end{bmatrix} \\ &= \begin{bmatrix} \varepsilon & 6 \otimes u_2(k) \otimes \overline{u_3(k)} & 6 \otimes \overline{u_1(k)} \otimes \overline{u_2(k)} \\ 6 \otimes \overline{u_2(k)} \otimes u_3(k) & \varepsilon & 6 \otimes u_1(k) \otimes \overline{u_3(k)} \\ 6 \otimes u_1(k) \otimes u_2(k) & 6 \otimes \overline{u_1(k)} \otimes u_3(k) & \varepsilon \end{bmatrix} \end{aligned}$$

The diagonal elements in $A^{\otimes^2}_{0,4,\mathrm{d}}(u(k),k)$ are ε by definition since $\overline{u_i(k)} \otimes u_i(k) = \varepsilon$ and $\varepsilon \otimes a = \varepsilon$. The next matrix power of $A_{0,4,\mathrm{d}}$ is:

$$A^{\otimes^3}_{0,4,\mathrm{d}}(u(k),k) = \begin{bmatrix} 9 \otimes (u_1(k) \otimes u_2(k) \otimes \overline{u_3(k)} \oplus \overline{u_1(k)} \otimes \overline{u_2(k)} \otimes u_3(k)) & & \\ 9 \otimes u_1(k) \otimes \underline{u_2(k)} \otimes \overline{u_2(k)} & \cdots & \\ 9 \otimes u_2(k) \otimes \overline{u_2(k)} \otimes u_3(k) & & \end{bmatrix}$$

$$\begin{bmatrix} 9 \otimes \overline{u_1(k)} \otimes u_3(k) \otimes \overline{u_3(k)} \\ 9 \otimes (u_1(k) \otimes u_2(k) \otimes \overline{u_3(k)} \oplus \overline{u_1(k)} \otimes \overline{u_2(k)} \otimes u_3(k)) \ \ldots \\ 9 \otimes u_2(k) \otimes u_3(k) \otimes \overline{u_3(k)} \end{bmatrix}$$

$$\begin{bmatrix} 9 \otimes u_1(k) \otimes \overline{u_1(k)} \otimes \overline{u_3(k)} \\ 9 \otimes u_1(k) \otimes \overline{u_1(k)} \otimes \overline{u_2(k)} \\ 9 \otimes (u_1(k) \otimes u_2(k) \otimes \overline{u_3(k)} \oplus \overline{u_1(k)} \otimes \overline{u_2(k)} \otimes u_3(k)) \end{bmatrix}$$

All non-diagonal elements are ε by definition. The diagonal elements are positive if $u_1(k) \otimes u_2(k) \otimes \overline{u_3(k)} = 0$ or $\overline{u_1(k)} \otimes \overline{u_2(k)} \otimes u_3(k) = 0$. That means these combinations of control variables correspond to infeasible train orders. So by simply determining the matrix powers, which has to be done to determine the explicit model anyway, and looking at the diagonal elements the infeasible combinations of control variables can be found. These combinations of control variables can be removed by replacing $u_1(k) \otimes u_2(k) \otimes \overline{u_3(k)}$ and $\overline{u_1(k)} \otimes \overline{u_2(k)} \otimes u_3(k)$ with ε.

To ensure the MILP does not use this combination of control variables constraints corresponding to the following max-plus-linear inequalities are added to the explicit MILP problem formulation:

$$\overline{u_1(k)} \otimes \overline{u_2(k)} \otimes u_3(k) \leq \varepsilon$$
$$u_1(k) \otimes u_2(k) \otimes \overline{u_3(k)} \leq \varepsilon$$

A.3 Switching Between Tracks

As an example consider two parallel tracks, track 1 and 2, with two trains on each of the tracks. Trains 1 and 2 traverse track 1 and trains 3 and 4 traverse track 2. All trains are running in the same cycle. The timetable period is 1 hour, and the timetable vector is $r(0) = \begin{bmatrix}0 & 30 & 5 & 35 & 15 & 50 & 20 & 55\end{bmatrix}^\top$. We can define the following sets and vectors based on this example:

$$T_1 = \begin{bmatrix}1 & 2\end{bmatrix}^\top \qquad T_2 = \begin{bmatrix}3 & 4\end{bmatrix}^\top \qquad \tilde{T}_1 = \begin{bmatrix}1 & 2 & 3 & 4\end{bmatrix}^\top$$
$$\mathcal{H}_1 = \emptyset \qquad \mathcal{H}_2 = \{1\} \qquad \mathcal{H}_3 = \emptyset \qquad \mathcal{H}_4 = \{3\}$$

Using these sets, vectors and (47) we can determine the entries of $\tilde{A}_{\mu,4,\mathrm{d},1}$. Since all trains are in the same cycle only $\mu = 0$ needs to be considered. The first if-condition of (47) results in the following non-ε elements of $\tilde{A}_{0,4,\mathrm{d},1}$:

$$[\tilde{A}_{0,4,\mathrm{d},1}]_{1,2} = \tau_{\mathrm{h,d},1,2}(k) \otimes u_{1,2}(k) \otimes \big(w_1(k) \otimes w_2(k) \oplus \overline{w_1(k)} \otimes \overline{w_2(k)}\big)$$
$$[\tilde{A}_{0,4,\mathrm{d},1}]_{3,4} = \tau_{\mathrm{h,d},3,4}(k) \otimes u_{3,4}(k) \otimes \big(w_3(k) \otimes w_4(k) \oplus \overline{w_3(k)} \otimes \overline{w_4(k)}\big)$$

and the second if-condition results in the following non-ε elements of $\tilde{A}_{0,4,\mathrm{d},1}$:

$$[\tilde{A}_{0,4,\mathrm{d},1}]_{2,1} = \tau_{\mathrm{h,d},2,1}(k) \otimes \overline{u_{1,2}(k)} \otimes \big(w_1(k) \otimes w_2(k) \oplus \overline{w_1(k)} \otimes \overline{w_2(k)}\big)$$
$$[\tilde{A}_{0,4,\mathrm{d},1}]_{4,3} = \tau_{\mathrm{h,d},4,3}(k) \otimes \overline{u_{3,4}(k)} \otimes \big(w_3(k) \otimes w_4(k) \oplus \overline{w_3(k)} \otimes \overline{w_4(k)}\big).$$

The first two if-conditions result in the headway times between departure events of the trains, that traverse the same track during nominal operation, with the addition of the control variables for track selection. The third if-condition results in the following non-ε elements $\tilde{A}_{0,4,\mathrm{d},1}$:

$$[\tilde{A}_{0,4,\mathrm{d},1}]_{1,3} = \tau_{\mathrm{h,d},1,3}(k) \otimes u_{1,3}(k) \otimes \big(w_1(k) \otimes \overline{w_3(k)} \oplus \overline{w_1(k)} \otimes w_3(k)\big)$$
$$[\tilde{A}_{0,4,\mathrm{d},1}]_{1,4} = \tau_{\mathrm{h,d},1,4}(k) \otimes u_{1,4}(k) \otimes \big(w_1(k) \otimes \overline{w_4(k)} \oplus \overline{w_1(k)} \otimes w_4(k)\big)$$
$$[\tilde{A}_{0,4,\mathrm{d},1}]_{2,4} = \tau_{\mathrm{h,d},2,4}(k) \otimes u_{2,4}(k) \otimes \big(w_2(k) \otimes \overline{w_4(k)} \oplus \overline{w_2(k)} \otimes w_4(k)\big)$$
$$[\tilde{A}_{0,4,\mathrm{d},1}]_{3,2} = \tau_{\mathrm{h,d},3,2}(k) \otimes u_{3,2}(k) \otimes \big(w_3(k) \otimes \overline{w_2(k)} \oplus \overline{w_3(k)} \otimes w_2(k)\big),$$

and the fourth if-condition results in the final non-ε elements $\tilde{A}_{0,4,\mathrm{d},1}$:

$$
\begin{aligned}
[\tilde{A}_{0,4,\mathrm{d},1}]_{3,1} &= \tau_{\mathrm{h,d},3,1}(k) \otimes \overline{u_{1,3}(k)} \otimes \big(w_1(k) \otimes \overline{w_3(k)} \oplus \overline{w_1(k)} \otimes w_3(k)\big) \\
[\tilde{A}_{0,4,\mathrm{d},1}]_{4,1} &= \tau_{\mathrm{h,d},4,1}(k) \otimes \overline{u_{1,4}(k)} \otimes \big(w_1(k) \otimes \overline{w_4(k)} \oplus \overline{w_1(k)} \otimes w_4(k)\big) \\
[\tilde{A}_{0,4,\mathrm{d},1}]_{4,2} &= \tau_{\mathrm{h,d},4,2}(k) \otimes \overline{u_{2,4}(k)} \otimes \big(w_2(k) \otimes \overline{w_4(k)} \oplus \overline{w_2(k)} \otimes w_4(k)\big) \\
[\tilde{A}_{0,4,\mathrm{d},1}]_{2,3} &= \tau_{\mathrm{h,d},2,3}(k) \otimes \overline{u_{3,2}(k)} \otimes \big(w_3(k) \otimes \overline{w_2(k)} \oplus \overline{w_3(k)} \otimes w_2(k)\big).
\end{aligned}
$$

These headway times define the order of the trains if a train from track 1 switches to track 2 or the other way around. For the headway times between arrival events only $\tau_{\mathrm{h,d},i,j}(k)$ needs to be replaced with $\tau_{\mathrm{h,a},i,j}(k)$.

A.4 Matrix Multiplication Using Graph Theory

Consider the matrix A_0

$$
A_0 = \begin{bmatrix} A_\mathrm{a} & A_\mathrm{b} \\ A_\mathrm{c} & A_\mathrm{d} \end{bmatrix},
$$

and its graph $\mathcal{G}(A_0)$ given in Figure 10. According to graph theory $[A^{\otimes^m}]_{i,j}$ cor-

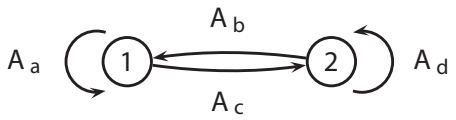

Fig. 10 Graph $\mathcal{G}(A_0)$

responds to the maximum weight of all paths of length m from node j to node i in the precedence graph of A. If we apply this to matrix A_0 for element $[A_0^{\otimes^2}]_{1,1}$, then we should look at all paths of length 2 in the graph that start and end at node 1. There are two paths of length 2 from node 1 to node 1. One consists of taking the edge from node 1 to node 1 twice resulting in a weight of $A_\mathrm{a}^{\otimes 2}$. The other path of length two consists of taking the edge from node 1 to node 2 and the edge from node 2 to node, resulting in a path weight of $A_\mathrm{b} \otimes A_\mathrm{c}$. The value of $[A_0^{\otimes^2}]_{1,1}$ is the maximum of the weight of both paths: $[A_0^{\otimes^2}]_{1,1} = A_\mathrm{a}^{\otimes 2} \oplus A_\mathrm{b} \otimes A_\mathrm{c}$. For the other elements of $A_\mathrm{a}^{\otimes 2}$ this results in:

$$
\begin{aligned}
[A_0^{\otimes^2}]_{1,2} &= A_\mathrm{a} \otimes A_\mathrm{b} \oplus A_\mathrm{b} \otimes A_\mathrm{d} \\
[A_0^{\otimes^2}]_{2,1} &= A_\mathrm{c} \otimes A_\mathrm{a} \oplus A_\mathrm{d} \otimes A_\mathrm{c} \\
[A_0^{\otimes^2}]_{2,2} &= A_\mathrm{d}^{\otimes 2} \oplus A_\mathrm{c} \otimes A_\mathrm{b}
\end{aligned}
$$

For the elements of $A_0^{\otimes^3}$ we should look at all paths of length 3. There are four path of length 3 starting at node 1 and ending at node 1

- Take the edge from node 1 to node 1 three times. This path has a weight of $A_\mathrm{a}^{\otimes 3}$.

- Take the edge from node 1 to node 1 a single time and then go to node 2 and back. This path has a weight of $A_\mathrm{a} \otimes A_\mathrm{b} \otimes A_\mathrm{c}$.
- First go to node 2 and back, then take the edge from 1 node to node 1. This path has a weight of $A_\mathrm{b} \otimes A_\mathrm{c} \otimes A_\mathrm{a}$.
- Take the edge from node 1 to node 2, then the edge from node 2 to node 2, and finally the edge from node 2 to node 1. This path has a weight of $A_\mathrm{b} \otimes A_\mathrm{d} \otimes A_\mathrm{c}$.

This can also be done for the other elements and for all elements of all other matrix powers. By looking at the graph of A_0 and the relation between paths in the graph and elements of the matrix powers of the matrix, it is clear only a limited number of paths is possible and that there is a clear structure in these paths. For paths starting and ending in node 1, if the edge from node 1 to node 2 is taken, the edge from node 2 to node 1 must also be in the path. If the edge from node 2 to node 2 is in the path then the edges from node 1 to node 2 and from node 2 to node 1 must also be in the path. The edge from node 1 to node 1 can only be in the path before the edge from node 1 to node 2, after the edge from node 2 to node 1, and before and after itself. And the number of edges is equal to the matrix power. Such rules can be set up for all four of the elements and result in (59–62) in Section 4.

References

Baccelli F, Cohen G, Olsder G, Quadrat J (1992) Synchronization and Linearity: An Algebra for Discrete Event Systems. Wiley, New York

Braker JG (1991) Max-algebra modelling and analysis of time-dependent transportation networks. In: Proceedings of the 1st European Control Conference, Grenoble, France, pp 1831–1836

Braker JG (1993) An extended algorithm for performance evaluation of timed event graphs. In: Proceedings of the 2nd European Control Conference, Groningen, The Netherlands, pp 524–529

Caimi G, Fuchsberger M, Laumanns M, Lüthi M (2012) A model predictive control approach for discrete-time rescheduling in complex central railway station areas. Computers and Operations Research 39(11):2578–2593

Corman F, D'Ariano A, Pacciarelli D, Pranzo M (2012) Bi-objective conflict detection and resolution in railway traffic management. Transportation Research Part C: Emerging Technologies 20(1):79–94

Cuninghame-Green R (1979) Minimax Algebra, Lecture Notes in Economics and Mathematical Systems, vol 166. Springer-Verlag, Berlin, Germany

D'Ariano A, Pacciarelli D, Pranzo M (2007) A branch and bound algorithm for scheduling trains in a railway network. European Journal of Operational Research 183(2):643–657

De Schutter B, van den Boom T, Hegyi A (2002) A model predictive control approach for recovery from delays in railway systems. Transportation Research Record 1793:15–20

GLPK (2014) Gnu linear programming kit. URL `http://www.gnu.org/software/glpk/`

Goverde RMP (2007) Railway timetable stability analysis using max-plus system theory. Transportation Research Part B: Methodological 41(2):179 – 201

Goverde RMP (2010) A delay propagation algorithm for large-scale railway traffic networks. Transportation Research Part C: Emerging Technologies 18(3):269 – 287

Gurobi (2014) Gurobi optimizer reference manual. URL `http://www.gurobi.com`

Hansen IA, Pachl J (2008) Railway Timetable & Traffic: Analysis - Modelling - Simulation. Eurailpress, Hamburg, Germany

Heidergott B, Vries R (2001) Towards a (max,+) control theory for public transportation networks. Discrete Event Dynamic Systems 11(4):371–398

Heidergott B, Olsder GJ, van der Woude J (2006) Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications. Princeton Series in Applied Mathematics, Princeton University Press, Princeton, United States of America

Kecman P, Corman F, D'Ariano A, Goverde RM (2013) Rescheduling models for railway traffic management in large-scale networks. Public Transport 5(1-2):95–123

Kersbergen B, van den Boom TJJ, De Schutter B (2013a) On implicit versus explicit max-plus modeling for the rescheduling of trains. In: 5th International Seminar on Railway Operations Modelling and Analysis (RailCopenhagen), Copenhagen, Denmark

Kersbergen B, van den Boom TJJ, De Schutter B (2013b) Reducing the time needed to solve the global rescheduling problem for railway networks. In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC2013), The Hague, The Netherlands

Kroon LG, Peeters LWP (2003) A variable trip time model for cyclic railway timetabling. Transportation Science 37(2):198–212

Minciardi R, Paolucci M, Pesenti R (1995) Generating optimal schedules for an underground railway line. In: Proceedings of the 34th IEEE Conference on Decision and Control, New Orleans, Louisiana, pp 4082–4085

TOMLAB (2014) Tomlab optimization environment. URL `http://tomopt.com/tomlab/`

Törnquist Krasemann J (2012) Design of an effective algorithm for fast response to the rescheduling of railway traffic during disturbances. Transportation Research Part C: Emerging Technologies 20(1):62–78

van den Boom TJJ, De Schutter B (2006) Modelling and control of discrete event systems using switching max-plus-linear systems. Control Engineering Practice 14(10):1199–1211

van den Boom TJJ, Weiss N, Leune W, Goverde RMP, De Schutter B (2011) A permutation-based algorithm to optimally reschedule trains in a railway traffic network. In: Proceedings of the 18th IFAC World Congress, Milan, Italy, pp 9537–9542

van den Boom TJJ, Kersbergen B, De Schutter B (2012) Structured modeling, analysis, and control of complex railway operations. In: Proceedings of the 51st IEEE Conference on Decision and Control, Maui, Hawaii, pp 7366–7371

de Vries R, De Schutter B, De Moor B (1998) On max-algebraic models for transportation networks. In: Proceedings of the 4th International Workshop on Discrete Event Systems (WODES'98), Cagliari, Italy, pp 457–462

de Waal P, Overkamp A, van Schuppen JH (1997) Control of railway traffic on a single line. In: Proceedings of the European Control Conference (ECC'97), Brussels, Belgium, paper 230