

Technical report 13-032

Reducing the Time Needed to Solve the Global Rescheduling Problem for Railway Networks*

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

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

B. Kersbergen, T. van den Boom, and B. De Schutter, “Reducing the time needed to solve the global rescheduling problem for railway networks,” *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, The Netherlands, pp. 791–796, Oct. 2013.

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://dpub.eu/13-032>

Reducing the time needed to solve the global rescheduling problem for railway networks*

Bart Kersbergen¹, Ton van den Boom¹ and Bart De Schutter¹

Abstract—In this paper a method is introduced to reduce the computation time needed to solve the global rescheduling problem for railway networks. The railway network is modeled as a switching max-plus-linear model. This model is used to determine the constraints of the rescheduling problem. The rescheduling problem is described as a Mixed Integer Linear Programming (MILP) problem. The dispatching actions in this implementation are limited to changing the order of the trains and breaking connections at stations. These dispatching actions are most effective for smaller delays. It is therefore assumed that the delays are less than some maximum value. The proposed reduction method determines which (combinations of) control inputs will never be used if the delays are below this maximum value and removes them, as well as the constraints associated to them, resulting in a smaller model. Using the reduced model in the MILP problem significantly decreases the time needed to solve the MILP problem while still yielding the optimal solution for the original MILP problem.

I. INTRODUCTION

During the day to day operation of a railway network many small delays may occur. If these delays are not dealt with appropriately, they can propagate through large parts of the network and cause even more delays. In recent years researchers have been developing systems to support the dispatchers in their task. Most of these systems are limited to solving local dispatching problems [1] and [2]. The main reason why many researchers only consider a part of the network is the complexity of the problem. Many implementations involve the solution of a Mixed Integer Linear Programming (MILP) problem. The (worst-case) time needed to solve an MILP problem is generally assumed to grow exponentially with the size of the problem. Therefore, a MILP based on a part of the network is much easier to solve than one based on the entire network. Several researchers have also been working on solving the dispatching problem for the entire network [3] and [4]. One of the biggest challenges of solving the global dispatching problem is to find the optimal solution quickly. In this paper we propose a method to reduce the time needed to solve the problem, by limiting the possible control actions.

We build on the work of [4], [5] and [6], where an explicit Switching Max-Plus-Linear (SMPL) model of the railway traffic is introduced. This model is used in a rescheduling problem that determines the optimal order of trains for

the entire network [5] and [6]. We continue this work by introducing a method to reduce the computation time needed to solve the rescheduling problem by limiting the control freedom.

II. MODELING

The operation of the railway network can be split into two different modes. When there are no delays all trains run according to the predetermined routes and arrive and depart according to the timetable. This is called the *nominal operation*. If delays occur, some trains will not depart and arrive according to the timetable and rescheduling actions may have to be taken to limit the propagation of these delays; this is called the *perturbed operation*. First the model of the railway traffic is presented while it is running according to the nominal operation. This will then be extended to include rescheduling such that the perturbed operation can also be modeled.

A. Nominal Operation

The model of [4], [5] and [6] is based on a periodic timetable, because in many countries the passenger railways operate according to one. The railway traffic is modeled as a cyclic discrete-event system. The arrivals and departures of the trains, at all stations and junctions outside the interlocking areas of the stations, are the events of the system. Stations and their interlocking areas are modeled as single points with unlimited capacity, junctions outside the interlocking area of a station are modeled as single points with limited capacity (only one train can be on a junction at the same time). Tracks between stations and junctions are modeled as single links; no block sections are considered. Instead of a signaling system, headway times are used to determine the order of trains and to separate trains running over the same track. In one cycle all arrival and departure events of one timetable period are modeled. The current cycle is denoted by k and it is assumed all events of past cycles have already occurred, therefore the event times of events of past cycles are fixed. The timetable period is denoted by T .

If the prediction horizon consists of several periods of the timetable, then the timetable must be extended to cover the whole prediction horizon.

The model of the railway traffic is built up from *train runs*. We define a train run as the following combination of actions: a train departs from a station or junction, it drives over a track, and arrives at the next station or junction. Each train run has an index i , and an associated arrival time a_i and departure time d_i . Each event time has a cycle counter

*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”.

¹B. Kersbergen, T.J.J. van den Boom and B. De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, The Netherlands.

k , which denotes the cycle the event time is in. The relation between the arrival time and departure time of a train run can be described by a *running time* constraint, defined for train run i as:

$$a_i(k) \geq d_i(k) + \tau_i^{\text{run}}(k), \quad (1)$$

where $\tau_i^{\text{run}}(k)$ is the time the train needs to traverse the track.

A single train moving through the network can be described by a sequence of train runs connected to each other through *continuity* constraints. Train runs i and j of the same physical train, where train run i starts after train run j has completed, are connected through the following continuity constraint:

$$d_i(k) \geq a_j(k - \mu_{ij}) + \tau_{ij}^{\text{dwell}}(k) \quad (2)$$

where $\tau_{ij}^{\text{dwell}}(k)$ is the time the trains needs to wait at the station for the passengers to board and alight, or at line ends it is the time for passengers to board and alight plus the time needed to turn the train around, and μ_{ij} is zero if the arrival and departure events are in the same cycle and one if there is one cycle difference between them.

Since the railway network operates 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. The timetable constraints are described by:

$$d_i(k) \geq r_i^{\text{d}}(k) = r_i^{\text{d}}(0) + T \times k \quad (3)$$

$$a_i(k) \geq r_i^{\text{a}}(k) = r_i^{\text{a}}(0) + T \times k, \quad (4)$$

where $r_i^{\text{d}}(0)$ and $r_i^{\text{a}}(0)$ are the scheduled departure and arrival time of train run i for the first period that is modeled. In many countries trains are allowed to arrive before their scheduled arrival time; in that case the timetable constraint on the arrival time (4) may be left out for all train runs.

The order of the trains on the tracks and the minimum distance needed between the trains is modeled by *headway* constraints. For train run i , define the set \mathcal{H}_i as the set of train runs that share the same track as train run i and start before train run i and for which the trains traverse the track in the same direction. Define \mathcal{S}_i as the set of train runs on the same track that start before train run i and traverse the track in opposite direction. The headway constraints for train run i are defined as

$$d_i(k) \geq d_l(k - \mu_{il}) + \tau_{il,\text{d}}^{\text{headway}}(k) \quad (5)$$

$$a_i(k) \geq a_l(k - \mu_{il}) + \tau_{il,\text{a}}^{\text{headway}}(k), \quad (6)$$

for each $l \in \mathcal{H}_i$, where $\tau_{il,\text{d}}^{\text{headway}}(k)$ is the headway time needed between the departures of the two trains, $\tau_{il,\text{a}}^{\text{headway}}(k)$ is the headway time needed between the arrivals of the trains, and where μ_{il} is defined in the same way as for (2), and

$$d_i(k) \geq a_m(k - \mu_{im}) + \tau_{im}^{\text{sep}}(k), \quad (7)$$

for each $m \in \mathcal{S}_i$, where $\tau_{im}^{\text{sep}}(k)$ is the separation time between the two trains, and μ_{im} is defined in the same way as for (2).

At some stations passengers can transfer to other trains. Transfers that are guaranteed by the railway operators, are modeled by *connection* constraints. Define \mathcal{C}_i as the set of train runs to which train run i has to give a connection to. Then the connection constraints for train run i are defined as:

$$d_i(k) \geq a_c(k - \mu_{ic}) + \tau_{ic}^{\text{connect}}(k), \quad (8)$$

for each $c \in \mathcal{C}_i$, where $\tau_{ic}^{\text{connect}}(k)$ is the time needed for the passengers to transfer from the train of train run c to the train of train run i . During nominal operation all process times (running, dwell, headway, separation, and connection times) in the model are set to the minimum times needed to complete the processes.

Most events will have a combination of these constraints. For every event all of these constraints can be gathered into a single equation and described as a linear function in max-plus algebra. 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 [7]:

$$a \oplus b = \max(a, b)$$

$$a \otimes b = a + b,$$

for $a, b \in \mathbb{R}_\varepsilon$.

For matrices these operators are defined as:

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})$$

$$[A \otimes C]_{ij} = \bigoplus_{k=1}^n a_{ik} \otimes c_{kj} = \max_{k=1, \dots, n} (a_{ik} + c_{kj}),$$

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

By assuming that trains depart and arrive as soon as all constraints are satisfied, (1) through (8) can be written as two max-plus-linear¹ equations [6]:

$$\begin{aligned} d_i(k) = & a_j(k - \mu_{ij}) \otimes \tau_{ij}^{\text{dwell}}(k) \oplus \\ & \bigoplus_{l \in \mathcal{H}_i} \left(d_l(k - \mu_{il}) \otimes \tau_{il,\text{d}}^{\text{headway}}(k) \right) \oplus \\ & \bigoplus_{m \in \mathcal{S}_i} \left(a_m(k - \mu_{im}) \otimes \tau_{im}^{\text{sep}}(k) \right) \oplus \\ & \bigoplus_{c \in \mathcal{C}_i} \left(a_c(k - \mu_{ic}) \otimes \tau_{ic}^{\text{connect}}(k) \right) \oplus r_i^{\text{d}}(k) \end{aligned} \quad (9)$$

$$\begin{aligned} a_i(k) = & \bigoplus_{l \in \mathcal{H}_i} \left(a_l(k - \mu_{il}) \otimes \tau_{il,\text{a}}^{\text{headway}}(k) \right) \oplus \\ & d_i(k) \otimes \tau_i^{\text{run}}(k) \oplus r_i^{\text{a}}(k). \end{aligned} \quad (10)$$

By determining these max-plus-linear equations for all d_i and a_i and collecting them in a state vector $x(k)$ the model can be written as a max-plus-linear model defined as:

$$x(k) = A_0(k) \otimes x(k) \oplus A_1(k) \otimes x(k-1) \oplus r(k), \quad (11)$$

¹A max-plus-linear equation is an equation that is linear in max-plus algebra; it has the following form: $y = a \otimes x \oplus b$.

where $A_0(k), A_1(k) \in \mathbb{R}_\varepsilon^{2q \times 2q}$ and $x(k)$ and $r(k)$ are defined as:

$$\begin{aligned} x(k) &= [d_1(k) \ \dots \ d_q(k) \ a_1(k) \ \dots \ a_q(k)]^\top, \\ r(k) &= [r_1^d(k) \ \dots \ r_q^d(k) \ r_1^a(k) \ \dots \ r_q^a(k)]^\top, \end{aligned}$$

where q is the number of train runs. The elements of the matrix $A_0(k)$ contain the process times and control inputs of the constraints between events in the same cycle ($\mu_{ij} = 0$) and the elements of the matrix $A_1(k)$ contain the process times and control inputs of the constraints between events of the current and the previous cycle ($\mu_{ij} = 1$). The entries of the matrices that do not correspond to any constraint are equal to ε .

B. Perturbed Operation

In [4] and [6], the model of the previous section has been extended to include delays and control inputs that allow for the reordering of the trains on the tracks. Delays are added to the model by increasing the process times of the processes that cause the delays. Since one cycle can model the events of several timetable periods and it is assumed all events of the previous cycle are in the past and therefore cannot be changed, the control variables are only added between events of the same cycle.

For train run i the headway constraints, for trains traversing the track in the same direction in the same cycle, become

$$d_i(k) \geq d_l(k) \otimes \tau_{il,d}^{\text{headway}}(k) \otimes u_{il}(k) \quad (12)$$

$$a_i(k) \geq a_l(k) \otimes \tau_{il,a}^{\text{headway}}(k) \otimes u_{il}(k) \quad (13)$$

$$d_l(k) \geq d_i(k) \otimes \tau_{li,d}^{\text{headway}}(k) \otimes \overline{u_{il}}(k) \quad (14)$$

$$a_l(k) \geq a_i(k) \otimes \tau_{li,a}^{\text{headway}}(k) \otimes \overline{u_{il}}(k), \quad (15)$$

for each $l \in \mathcal{H}_i$, where $u_{il}(k) \in \{\varepsilon, 0\}$ is the control variable with $\overline{u_{il}}(k) = 0$ if $u_{il}(k) = \varepsilon$, and $\overline{u_{il}}(k) = \varepsilon$ if $u_{il}(k) = 0$. For $u_{il}(k) = 0$, (12) and (13) are the default headway constraints and define the order of train runs: "l before i", and (14) and (15) are always valid, since all event times are larger than $\varepsilon = -\infty$. If $u_{il}(k) = \varepsilon$, (12) and (13) are always valid, and (14) and (15) define a different order of train runs: "i before l".

The same principle is applied to the headway constraints of the trains traversing the track in the opposite direction in the same cycle:

$$d_i(k) \geq a_m(k) \otimes \tau_{im}^{\text{sep}(k)} \otimes u_{im}(k) \quad (16)$$

$$d_m(k) \geq a_i(k) \otimes \tau_{mi}^{\text{sep}(k)} \otimes \overline{u_{im}}(k) \quad (17)$$

for each $m \in \mathcal{S}_i$.

By adding control variables to the headway constraints of all train runs for which the order can be changed, we can model the effects of reordering trains in the network at certain points. The resulting model can be described as:

$$\begin{aligned} x(k) &= A_0(u(k), k) \otimes x(k) \oplus \\ &A_1(k) \otimes x(k-1) \oplus r(k), \end{aligned} \quad (18)$$

where $u(k)$ is the set of control variables containing all $u_{il}(k)$ and $\overline{u_{il}}(k)$, and the elements of $A_0(u(k), k)$ are max-plus-linear functions in the control variables. This model is called a Switching Max-Plus-Linear (SMPL) model, since it can switch between behaviors (train orders) and each behavior is described by a max-plus-linear model.

C. Explicit model

The model of (18) is called an implicit model because the state vector $x(k)$ does not only depend on the state vector of the previous cycle (and the timetable reference), but also on itself. In [6] this model is rewritten into its explicit form. In this way the dependency of $x(k)$ on itself is removed.

The implicit max-plus-linear model in 18 can be rewritten into its explicit form by determining $A_0^*(u(k), k)$.

In general A^* can be calculated for any $A \in \mathbb{R}_\varepsilon^{m \times n}$, see for instance [7], by

$$A^* = \bigoplus_{p=0}^{\infty} A^{\otimes p}, \quad (19)$$

with $A^{\otimes p} = A^{\otimes p-1}$ for $p \leq 1$, and $A^{\otimes 0} = E$, where E is the max-plus identity matrix; this is a square matrix with diagonal entries equal to 0 and the rest of its entries ε . Matrix A^* only exists if there are no circuits of positive weight in the weighted graph of A [7].

The explicit model resulting from the implicit max-plus-linear model in (18) can be written as

$$\begin{aligned} x(k) &= A_0^*(u(k), k) \otimes A_1(k) \otimes x(k-1) \oplus \\ &A_0^*(u(k), k) \otimes r(k) \end{aligned} \quad (20)$$

This model is not an ordinary max-plus-linear model but an SMPL model, containing infeasible train orders, resulting in infinite event times. These infeasible train orders correspond to circuits of positive weight in the weighted graph of $A_0(u(k), k)$, as a result $A_0^*(u(k), k)$ cannot be calculated. By finding all the circuits in the weighted graph of $A_0(u(k), k)$, the combinations of control inputs corresponding to the infeasible train orders can be identified. By removing these combinations of control inputs, by replacing any element of the max-plus matrix powers of $A_0(u(k), k)$ that contains one of these combination of control inputs by ε during the calculation of the explicit model, $A_0^{*,\text{feas}}(u(k), k)$ can be calculated. This matrix contains the part of $A_0^*(u(k), k)$ corresponding to the feasible train orders only. The exact details on how to determine $A_0^{*,\text{feas}}(u(k), k)$ can be found in [6]. The resulting explicit SMPL model can be written as:

$$\begin{aligned} x(k) &= A^{\text{exp}}(u(k), k) \otimes x(k-1) \oplus \\ &A_0^{*,\text{feas}}(u(k), k) \otimes r(k), \end{aligned} \quad (21)$$

where $A^{\text{exp}} = A_0^{*,\text{feas}}(u(k), k) \otimes A_1(k)$.

III. REDUCING THE COMPUTATION TIME

The reduction method that we will introduce in this section can be applied off-line during the calculation of the explicit model. To simplify the notation of the matrices in the rest of this paper, we will write $A_0(u(k), k)$ as A_0 and $A_1(k)$ as A_1 .

A. Delay model

The model described in the previous section allows for the reordering of trains at certain tracks. This includes orders that are not likely to ever happen, such as letting the last train on a track start before all other train on a track, since this would only happen if the last train would not have a delay and all other trains would have large delays.

If we assume there is an upper bound to the delays then some train orders, such as the one described above, will never be used and the (combinations of) control inputs associated to those train orders can be removed from the model without affecting the optimal solution of the rescheduling problem. To determine these (combinations of) control inputs, the model needs to be transformed, such that the state vector $x(k)$ no longer shows the arrival and departure times, but the delays. To do this the negative slack time should be defined. We will use the negative slack time definition as given by [8].

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

The negative slack times can be arranged in two matrices based on the value of μ_{ij} , which is either zero or one:

$$[A_0^d]_{ij} = [A_0]_{ij} - (r_i(0) - r_j(0)) \quad (22)$$

$$[A_1^d]_{ij} = [A_1]_{ij} - (r_i(0) - (r_j(0) - T)), \quad (23)$$

where we used

$$r(k) = r(0) + k \times T,$$

which is derived from (3) and (4). Note: the negative slack times are always less or equal to zero during nominal operation, since a positive value indicates a delay.

Using these matrices the model of (18) can be rewritten as

$$x^d(k) = A_0^d \otimes x^d(k) \oplus A_1^d \otimes x^d(k-1) \oplus \mathbf{0}, \quad (24)$$

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

In this model the elements of A_0^d and A_1^d represent the negative slack times between the events. The value of element $[A_0^d]_{ij}$ shows how much event $x_i(k)$ is delayed directly by $x_j(k)$ and the elements of any matrix power of A_0^d show how much events are delayed indirectly by each other. This means that all elements of A_0^d and A_1^d should be negative or zero during nominal operation, since a positive value indicates that there will be a delay. This model transformation is based on the concepts of slack time, realizability and structural delays as described by [8] and [9].

Using this model description the following theorem can be defined:

Theorem 1: The elements of the matrix powers of A_0^d give lower bounds to the delays caused by reordering the trains, if and only if, the used process times are the minimal process times.

Proof: By using the minimum process times, an element of A_0 and can be written as:

$$[A_0]_{ij} = t_{ij}^{\min} + \Delta_{ij} + v_{ij}(u(k)),$$

where t_{ij}^{\min} is a (sum of) minimum process time(s), Δ_{ij} is a positive value equal to the difference between the (sum of) actual process time(s) and the (sum of) minimum process time(s), and $v_{ij}(u(k))$ is a (sum of) max-plus control variable(s). If the element is independent of control inputs, $v_{ij}(u(k))$ is zero. The negative slack times for $\mu_{ij} = 0$ can be written as:

$$[A_0^d]_{ij} = t_{ij}^{\min} + \Delta_{ij} + v_{ij}(u(k)) - (r_i(0) - r_j(0)).$$

Since $(r_i(0) - r_j(0))$ is fixed by the timetable and t_{ij}^{\min} is as small as possible, a lower bound for the negative slack time is given for $\Delta_{ij} = 0$:

$$[A_0^d]_{ij} = t_{ij}^{\min} + v_{ij}(u(k)) - (r_i(0) - r_j(0)).$$

Since a positive value for $[A_0^d]_{ij}$ indicates a delay for x_i , for the $u(k)$, for which $v_{ij}(u(k)) = 0$, and the minimum process times are used, these values are minimum as well; they are a lower bound to the delay of x_i .

Any element of any power of A_0 can be written as

$$[A_0^{\otimes p}]_{ij} = \max_l (t_{ij,l}^{\min} + \Delta_{ij,l} + v_{ij,l}(u(k))),$$

where $t_{ij,l}^{\min}$ is a (sum of) minimum process time(s) of element l of the maximization, $\Delta_{ij,l}$ is a positive value equal to the difference between the (sum of) actual process time(s) and the (sum of) minimum process time(s) of element l of the maximization and $v_{ij,l}(u(k))$ is a (sum of) max-plus control variable(s) of element l of the maximization. If the element is independent of control inputs, $v_{ij,l}(u(k))$ is zero. The negative slack times for $\mu_{ij} = 0$ can be written as:

$$[A_0^d]_{ij}^{\otimes p} = \max_l (t_{ij,l}^{\min} + \Delta_{ij,l} + v_{ij,l}(u(k)) - r_i(0) + r_j(0)).$$

For each element of the maximization the lower bound can be found by setting $\Delta_{ij,l} = 0$. Each of these lower bounds is also a lower bound to the delay for x_i , for the $u(k)$, for which $v_{ij,l}(u(k)) = 0$. ■

Clearly the advantage of rewriting the model using (22) and (23) is that the value of the elements of the matrices, and their max-plus matrix powers, can be used to determine a lower bound to the delay of the associated events for the relevant (combinations of) control inputs.

B. Reduction method

If we assume that there is a maximum value for the delays, then we can remove those (combinations of) control inputs that would cause delays that are larger than the set maximum delay. These (combinations of) control inputs would have no effect on the solution of the rescheduling problem, so by removing them we can reduce the complexity of the rescheduling problem, which in turn will reduce the computation time needed to solve the rescheduling problem. These (combinations of) control inputs can be identified and

removed by setting $\Delta_{ij} = 0$ for all elements of A_0^d and determining the control inputs that cause one or more of the elements of the max-plus powers of A_0^d to have a value larger than the maximum delay. In Theorem 1 it was proven that the values of these elements correspond to a lower bound of the delay when the minimum process times were used and $\Delta_{ij} = 0$, therefore the (combinations of) control inputs associated to these elements can be removed without having an effect on the optimal solution of the rescheduling problem. All elements of (the max-plus matrix powers of) A_0^d containing these (combinations of) control inputs are set to ε , effectively removing those (combinations of) control inputs from the model. Because we need to determine the max-plus matrix powers during the calculation of $A_0^{*,d}$, as was shown in (19), we can apply the reduction method while we calculate the explicit model, resulting in a reduced version $A_0^{*,d,\text{red}}$ and a reduced explicit model:

$$x^d(k) = A^{\text{exp},d,\text{red}} \otimes x^d(k-1) \oplus A_0^{*,d,\text{red}} \otimes \mathbf{0}, \quad (25)$$

where $A^{\text{exp},d,\text{red}} = A_0^{*,d,\text{red}} \otimes A_1^d$.

Since combinations of control inputs are not explicitly modeled in the implicit model, but implied by the implicit model structure, combinations of control inputs cannot be removed from the implicit model. As a result the reduction method is far less effective for the implicit model than for the explicit model. Another way of looking at it, is that the reduction method not only removes elements from $A_0^{*,d}$, but also from all of its max-plus matrix powers, and therefore the reduction will be larger in the explicit model.

The resulting reduced explicit model can be used in the rescheduling problem. The rescheduling problem can be recast as a MILP problem as was shown for the implicit and explicit model in [6],

The calculation of the explicit model and applying the reduction method takes about a minute for small problems and up to a few hours for more realistic networks. Because of this it cannot be applied during the on-line optimization, but has to be done before hand. Beforehand the maximum delay is not known, so the reduced explicit model should be determined for several different values of the maximum delay and then during the on-line optimization the model best suited to the current situation should be chosen.

IV. CASE STUDIES

In this section we will consider two case studies; the first will be a case study of 50 delay scenarios, in which the maximum delay is small enough such that the optimal solutions of the rescheduling problem using the full implicit model and the rescheduling problem using the reduced explicit model are the same. The rescheduling problem using the full implicit model will be called rescheduling problem Imp from now on and the rescheduling problem using the reduced explicit model will be called rescheduling problem Exp.

In the second case study 20 scenarios are evaluated for which the solutions to rescheduling problem Imp differs from

the optimal solutions for rescheduling problem Exp because the maximum delay in these scenarios is larger than the assumed maximum delay we used while determining the reduced explicit model.

The model of the railway network that will be used is a large part of the Dutch railway network; the timetable of the year 2006 is used and only intercity and interregional trains are considered. We have chosen 15 minutes as the maximum delay used in the reduction method. The full explicit model will not be considered since it has already been shown in [6] that it is slower than the implicit model. The model has a period of one hour, and there are 163 trains running over the network in one hour, resulting in 381 train runs, 762 continuous variables, and 644 control inputs. The prediction horizon of the rescheduling problem is set to one hour. Problem Imp has 6150 constraints and 644 control inputs and problem Exp has 17204 constraints and 644 control inputs.

A. Case study 1

The computational performance of the rescheduling problem is evaluated for a set of 50 randomly generated scenarios. In these scenarios 10% of the train runs are delayed by a randomly chosen value, that is based on a truncated Weibull distribution with scale parameter 5 and shape parameter 0.8. The Weibull distribution is cut off at 9 minutes, otherwise there would be a chance that one or more of the delays is much larger than the assumed maximum of 15 minutes. In these 50 scenarios the optimal solution for both rescheduling problems is the same. The sum of delays in the uncontrolled network and optimally controlled network are shown in Figure 1. The average sum of delays for the uncontrolled model is 813 minutes. The average sum of delays of the optimal solutions is 712 minutes. The average reduction of the sum of delays is thus 101 minutes, which is an average reduction of the sum of delays of 12.5%. The computation times needed to solve the rescheduling problems for these scenarios are shown, on a logarithmic scale, in Figure 2. The average computation time to solve rescheduling problem Imp is 5.91 seconds. The average computation time needed to solve rescheduling problem Exp is 0.11 seconds, which is 51 times faster.

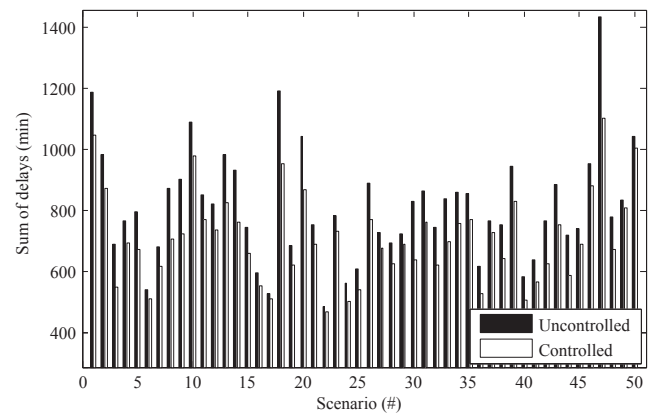


Fig. 1. Case study 1: Sum of delays for the 50 scenarios for the uncontrolled (black bars) and optimally controlled (white bars) case.

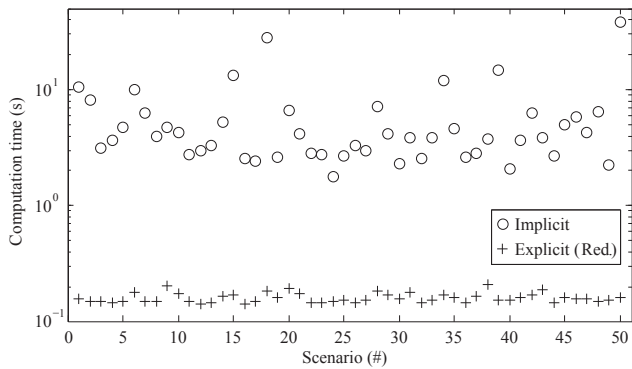


Fig. 2. Case study 1: Computation time of the solver for the 50 scenarios for the implicit (o) and reduced explicit (+) models.

B. Case study 2

In this case study the solution of rescheduling problem 1 compared to the solution of rescheduling problem 2. This is done for a set of 20 scenarios in which 10% of the trains are delayed by a randomly chosen value, that is determined on a truncated Weibull distribution with scale parameter 8 and shape parameter 0.8. The Weibull distribution is cut off at 10 minutes and the maximum delay in the uncontrolled network is more than 15 minutes. The maximum delays for these scenarios are between 18 and 31 minutes.

The sums of delays for the uncontrolled network and for the solutions found for both rescheduling problems are shown in Figure 3. The average sum of delays for the uncontrolled model is 1218 minutes, the average sum of delays of the optimal solution of rescheduling problem 1 is 1036 minutes and the average sum of delays of the optimal solution of rescheduling problem 2 is 1049 minutes. The reduction of the sum of delays for solution of rescheduling problem 1 is 182 minutes, while for the solution of rescheduling problem 2 is 169 minutes, which is 6,9% worse. In the worst case (scenario 1, with maximum delay 18 minutes), the reduction of the sum of delays for rescheduling problem 2 is 25% less than for rescheduling problem 1. For scenario 18, with the largest maximum delay of 31 minutes, the solutions found by solving both rescheduling problems result in nearly the same reduction of delays.

For 8 scenarios (scenarios 2, 10, 11, 12, 14, 15, 19 and 20) the solution found for rescheduling problem 2 matches that of rescheduling problem 1.

To be able to guarantee the quality of the found solution it is important that the maximum delay used in the reduction method is large enough. Determining reduced explicit models for different values of the maximum delays is necessary such that there are reduced models for delay scenarios with different maximum delays. As long as the maximum delay in the scenario is not larger than the assumed maximum delay, and small enough for the reduction method to be effective, the rescheduling problem based on the reduced explicit model can be solved quicker than the rescheduling problem based on the implicit model, while the solution will be the same.

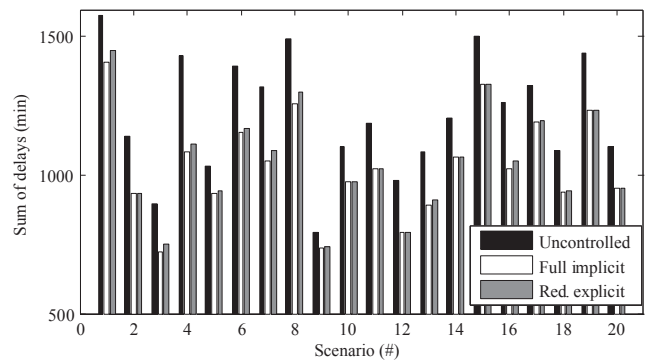


Fig. 3. Case study 2: Sum of delays for the 20 scenarios for the uncontrolled (black bars) network, implicit model (white bars) and the reduced explicit model (gray bars).

V. CONCLUSIONS

We have proposed a method to reduce the computation time needed to solve the rescheduling problem by limiting the possible control actions for the explicit switching max-plus-linear model introduced in [6]. The control actions that have no effect on the solution of the rescheduling problem are removed. As a result a model with reduced, but sufficient, rescheduling capabilities is derived.

In a case study we have compared the computation time needed to solve the rescheduling problem for the implicit model and the limited explicit model. From this case study it can be concluded that the optimal solution to the rescheduling problem using the limited explicit model is found much quicker, than for the rescheduling problem using the implicit model.

REFERENCES

- [1] A. D'Ariano and M. Pranzo, "An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances," *Networks and Spatial Economics*, vol. 9, no. 1, pp. 63–84, 2009.
- [2] J. T. Krasemann, "Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances," *Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 62–78, 2012.
- [3] P. Kecman, F. Corman, A. D'Ariano, and R. M. P. Goverde, "Rescheduling models for network-wide railway traffic management," in *Proceedings of the 12th Conference on Advanced Systems for Public Transport (CASPT 12)*, Santiago, Chile, July 2012, pp. 1–30.
- [4] T. J. J. van den Boom, B. Kersbergen, and B. De Schutter, "Structured modeling, analysis, and control of complex railway operations," in *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, Hawaii, Dec. 2012, pp. 7366–7371.
- [5] T. J. J. van den Boom, N. Weiss, W. Leune, R. M. P. Goverde, and B. De Schutter, "A permutation-based algorithm to optimally reschedule trains in a railway traffic network," in *Proceedings of the 18th IFAC World Congress*, Milan, Italy, Aug.–Sept. 2011, pp. 9537–9542.
- [6] B. Kersbergen, T. J. J. van den Boom, and B. De Schutter, "On implicit versus explicit max-plus modeling for the rescheduling of trains," in *Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis (RailCopenhagen)*, Copenhagen, Denmark, May 2013.
- [7] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat, *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Wiley, 1992.
- [8] R. M. P. Goverde, "Railway timetable stability analysis using max-plus system theory," *Transportation Research Part B: Methodological*, vol. 41, no. 2, pp. 179 – 201, 2007.
- [9] —, "A delay propagation algorithm for large-scale railway traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 3, pp. 269 – 287, 2010.