Delft University of Technology

Delft Center for Systems and Control

Technical report 13-015

On implicit versus explicit max-plus modeling for the rescheduling of trains*

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

If you want to cite this report, please use the following reference instead:
B. Kersbergen, T. van den Boom, and B. De Schutter, "On implicit versus explicit max-plus modeling for the rescheduling of trains," *Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis (RailCopenhagen)*, Copenhagen, Denmark, 15 pp., May 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

On Implicit versus Explicit Max-Plus Modeling for the Rescheduling of Trains

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

Abstract
In this paper a new railway traffic model is introduced. This model is determined by rewriting the model introduced in [12, 13]. The models are both macroscopic, contain the same level of details, allow the rescheduling of trains, and are formulated as Switching Max-Plus-Linear (SMPL) models, the difference being that the first model is an implicit SMPL model and the second one is an explicit SMPL model. In this paper a method is detailed for rewriting an implicit SMPL model into an explicit SMPL. Both models are used to solve the rescheduling problem. By solving the rescheduling problem a schedule for all trains in the network is found that minimizes the total delay. The rescheduling problem will be written as a Mixed Integer Linear Programming (MILP) problem and solved using the standard solver available in the Multi-Parametric Toolbox [11] for Matlab. The time needed to solve the rescheduling problem using either the implicit and explicit model is compared.

1 Introduction

In many countries railway traffic already covers a large part of the public transportation needs, while the number of passengers using trains is steadily increasing. Because of the increase of passengers the railway network is becoming more heavily utilized, resulting in timetables with less buffer times to compensate for delays. As a result, a small delay of a single train may cause numerous secondary delays and propagate through a large part of the network. Current practice of many railway operators is to divide the network into several dispatching areas, each with their own dispatcher. The dispatcher tries to reduce the number of secondary delays by taking dispatching actions such as rescheduling or rerouting trains, canceling trains, and breaking connections in his dispatching area. These actions are based on predetermined sets of rules and the experience of the dispatchers. As a result these actions may be optimal for the dispatching area, but may cause unnecessary and unforeseen delays in other parts of the network.

Rescheduling of railway traffic has been a topic of interest for many researchers in recent years [4, 5, 9, 10]. In [4, 5] the railway traffic is modeled as a microscopic model, while this gives the most accurate representation of the railway network and traffic, it also results in a very complex model. Only parts of the whole network are considered while solving the rescheduling problem, since solving the rescheduling problem for the whole network would take too much time to be practical for on-line use. In [9] it is shown that a less complex model, very similar to the model of [13], can be used for rescheduling. This model is described as an alternative graph and uses a solver specifically designed to minimize the maximum secondary delay. In [10] a greedy method is proposed that delivers good solutions in a matter of seconds. A greedy method is used because for some scenarios it takes too long to find the optimal scenario. The biggest problem that remains in all of the approaches is the computation time needed to find the optimal solution to the rescheduling problem for the entire railway network.

Railway traffic with fixed connections, predetermined routing, and a given schedule can be modeled using max-plus-linear models [2, 3, 6]. Max-plus algebra uses the two operators maximization and addition. These operators can be used on the set of real numbers together with minus infinity. For an extensive description of the properties and applications of max-plus algebra the reader is referred to [1]. More recently [13] has used Switching Max-Plus-Linear (SMPL) models to model railway traffic with adaptable order of trains on tracks for use in rescheduling. The advantages of using max-plus algebra to model the railway traffic are the tools and theories that have been developed for analyzing max-plus-linear models. The steady state behavior can be determined by finding the eigenvectors and values. Max-plus-linear models have been used for stability analyses of timetables [7] and delay propagation in railway networks [8]. All of these theories and methods are at our disposal when describing the model as a (switching) max-plus-linear model.

This paper continues the work of [12, 13], where a rescheduling method is introduced that uses a switching max-plus-linear (SMPL) model and determines the optimal control actions for the entire network in case of delays. We continue this work by introducing a modified description of the SMPL model in Section 2. This allows for the model to be rewritten into its explicit form, which is described in Section 3. In Section 4 the rescheduling problem is described for the implicit and explicit models and rewritten into a Mixed Integer Linear Programming (MILP) problem. The time needed to solve the rescheduling problem using the implicit and explicit model descriptions is compared for a case study of a small railway network in Section 5. In Section 6 the conclusions are drawn.

2 Modeling

In many countries the passenger railways operate on a periodic timetable. The reason for this is that a periodic timetable is easier to use and remember for the passengers. Therefore the model in this paper also uses a periodic timetable. The period of the timetable is denoted by T. During nominal operation all trains arrive and depart according to the timetable, the routes and connections of the trains are fixed and known, and the orders in which the trains occupy shared parts of the infrastructure are fixed. When trains are delayed, for example by longer boarding and alighting times of passengers, mechanical failures, speed restrictions due to signal failures or infrastructure problems, they may not be able to run according to the timetable and dispatchers will have to take action, such as reordering the trains at certain parts of the network, in order to reduce the effects of the initially delayed trains on other trains. In this case the railway traffic system is running in perturbed operation. First, the model of the nominal operation will be explained, and next it will be extended to the perturbed operation.

2.1 Nominal operation

The railway traffic is modeled as a discrete event system in [12, 13]. The events of the system are the arrivals and departures of the trains at all stations and all junctions outside the interlocking areas of the stations. All stations are modeled as single points with infinite capacity, tracks are modeled as single links, and headway times between trains running over the same tracks are modeled instead of the signaling system.

This results in a macroscopic model where the arrival and departure events are linked to each other through constraints based on the schedule of the trains and the infrastructure. These constraints can be characterized as one of the following types of constraints:

- Running time constraints

  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 finally arrives at the next station or junction. This is illustrated in Figure 1. A running time constraint then connects the departure time of train run i in cycle k, denoted by $d_i(k)$, to the arrival time at the next station or junction of the same train run denoted by $a_i(k)$ according to:

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

  where $\tau_i^{\text{run}}(k)$ is the *running time*, i.e., the time the train needs to drive over the track.

Figure 1: A train run

- Continuity constraints

  Since most trains have routes that are longer than a single train run, a constraint is needed to connect different train runs of the same train to each other. Consider train run i and the preceding train run j of the same train as shown in Figure 2. These train runs should be connected such that train run j does not start before train run i has ended at station B plus the time the train of train run j stays at the station, in order for the passengers to be able to board and alight, denoted by $\tau_{ij}^{\text{dwell}}(k)$. This is modeled by the continuity constraint described by:

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

  where μ_{ij} is zero if both train runs are in the same cycle and one if there is one cycle difference between the two train runs.

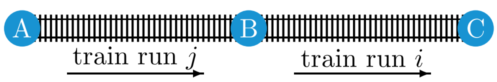

Figure 2: Train runs i and j of the same train

- Headway constraints

  Trains running over the same track cannot overtake each other on the track and should maintain a safe distance from each other. This can be modeled by adding a constraint that keeps these trains separated. This constraint is called the headway constraint. Headway constraints can be divided into two types: headway constraints for trains running in the same direction over the same track, and headway constraints for trains running in the opposite direction over the same track, as shown in Figures 3 and 4 respectively.

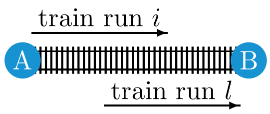

Figure 3: Train runs i and l on the same track in the same direction

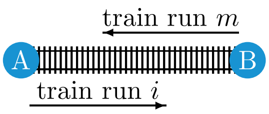

Figure 4: Train runs i and m on the same track in the opposite direction

  Define $\mathcal{H}_i$ as the set of train runs that end before train run i, and for which the trains run over the same track and in the same direction as the train of train run i. Then the headway constraints for train run i are described by:

$$d_i(k) \geq d_l(k - \mu_{il}) + \tau_{il}^{\text{headway}}(k) \tag{3}$$

$$a_i(k) \geq a_l(k - \mu_{il}) + \tau_{il}^{\text{headway}}(k), \tag{4}$$

  for each $l \in \mathcal{H}_i$, where $\tau_{il}^{\text{headway}}(k)$ is the *headway time* between train run i and train run l and μ_{il} is defined as before.

  Define $\mathcal{W}_i$ as the set of train runs that end before train run i, and for which the trains run over the same track and in the opposite direction as the train of train run i. Then the headway constraints for train run i are described by:

$$d_i(k) \geq a_m(k - \mu_{im}) + \tau_{im}^{\text{wait}}(k), \tag{5}$$

  for each $m \in \mathcal{W}_i$, where $\tau_{im}^{\text{wait}}(k)$ is the *wait time* that train run i has to start after train run m has ended and μ_{im} is again defined as before.

- Timetable constraints

  As was mentioned in the beginning of this section, the trains run according to a periodic timetable. To model the requirement that trains do not depart or arrive too early *timetable constraints* are used. These constraints are described by:

$$d_i(k) \geq r_i^{\mathrm{d}}(k) \tag{6}$$
$$a_i(k) \geq r_i^{\mathrm{a}}(k), \tag{7}$$

where $r_i^{\mathrm{d}}(k)$ and $r_i^{\mathrm{a}}(k)$ are the departure and arrival times as scheduled in the timetable. If trains are allowed to arrive early equation (7) can be left out.

- Connection constraints

  If a railway operator guarantees connections between trains (so that passengers can transfer from one train to the other), then another type of constraint has to be defined: the connection constraint. Define $\mathcal{C}_i$ as the set of train runs, train run i has to give a connection to. Then the connection constraints for train run i will be

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

  for each $c \in \mathcal{C}_i$, where $\tau_{ic}^{\mathrm{connect}}(k)$ is the minimum *connection time* needed for the passengers to alight and board the other train and μ_{ic} is defined the same as before.

An event can have several of these constraints. The set of constraints for each event can be combined into a single equation resulting in the following equations for the departure time of train run i and the arrival time of train run i:

$$\begin{aligned} d_i(k) = \max \Big(& a_j(k - \mu_{ij}) + \tau_{ij}^{\mathrm{dwell}}(k), \max_{l \in \mathcal{H}_i} \Big(d_l(k - \mu_{il}) + \tau_{il,d}^{\mathrm{headway}}(k) \Big), \\ & \max_{m \in \mathcal{W}_i} \Big(a_m(k - \mu_{im}) + \tau_{im}^{\mathrm{wait}}(k) \Big), \max_{c \in \mathcal{C}_i} \Big(a_c(k - \mu_{ic}) + \tau_{ic}^{\mathrm{connect}}(k) \Big), r_i^{\mathrm{d}}(k) \Big) \end{aligned} \tag{9}$$

$$a_i(k) = \max \Big(\max_{l \in \mathcal{H}_i} \Big(a_l(k - \mu_{il}) + \tau_{il,a}^{\mathrm{headway}}(k) \Big), d_i(k) + \tau_i^{\mathrm{run}}(k), r_i^{\mathrm{a}}(k) \Big) \tag{10}$$

Clearly only two operators are needed for these equations: the maximization and the plus operator. An algebra exists that uses only these two operators, it is called the max-plus algebra. The max-plus algebra is an idempotent semi-ring, consisting of the set $\mathbb{R}_{\max} = \mathbb{R} \cup \{\varepsilon\}$, where $\varepsilon = -\infty$, equipped with the two operators $\oplus$ and $\otimes$, that are defined as follows [1]:

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

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

These operators can be extended to matrices:

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

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

With the use of max-plus algebra, equations (9), (10) can be rewritten as:

$$d_i(k) = a_j(k-\mu_{ij}) \otimes \tau_{ij}^{\text{dwell}}(k) \oplus \bigoplus_{l\in\mathcal{H}_i} \Big(d_l(k-\mu_{il}) \otimes \tau_{il}^{\text{headway}}(k)\Big) \oplus$$
$$\bigoplus_{m\in\mathcal{W}_i} \Big(a_m(k-\mu_{im}) \otimes \tau_{im}^{\text{wait}}(k)\Big) \oplus \bigoplus_{c\in\mathcal{C}_i} \Big(a_c(k-\mu_{ic}) \otimes \tau_{ic}^{\text{connect}}(k)\Big) \oplus r_i^{\text{d}}(k) \quad (11)$$

$$a_i(k) = \bigoplus_{l\in\mathcal{H}_i} \Big(a_l(k-\mu_{il}) \otimes \tau_{il}^{\text{headway}}(k)\Big) \oplus d_i(k) \otimes \tau_i^{\text{run}}(k) \oplus r_i^{\text{a}}(k) \quad (12)$$

By defining the state vector $x(k)$ and timetable vector $r(k)$ as

$$x(k) = \begin{bmatrix} d_1(k) \\ \vdots \\ d_q(k) \\ a_1(k) \\ \vdots \\ a_q(k) \end{bmatrix}, \qquad r(k) = \begin{bmatrix} r_1^{\text{d}}(k) \\ \vdots \\ r_q^{\text{d}}(k) \\ r_1^{\text{a}}(k) \\ \vdots \\ r_q^{\text{a}}(k) \end{bmatrix},$$

where q is the number of train runs in the model, and by defining $A_0(k), A_1(k) \in \mathbb{R}_{\max}^{2q\times 2q}$, equations (11), (12) can be written in matrix form as

$$x(k) = A_0(k) \otimes x(k) \oplus A_1(k) \otimes x(k-1) \oplus r(k), \quad (13)$$

where the entries of $A_0(k)$, contain the running, headway, wait, dwell and connection times of the constraints between events in the same cycle ($\mu_{ij} = 0$) and the entries of $A_1(k)$, contain the running, headway, wait, dwell and connection times of the constraints between events of the current and previous cycle ($\mu_{ij} = 1$). All entries $[A_0(k)]_{ij}$ for which no constraint between train runs i and j with $\mu_{ij} = 0$ is defined, are equal to ε. The same applies for all entries $[A_1(k)]_{ij}$ for which no constraint between train runs i and j with $\mu_{ij} = 1$ is defined.

2.2 Perturbed operation

With the model introduced in Section 2.1 it is impossible to change the order of the trains. This functionality can be added to the model by making the headway constraints adjustable to the order of the trains. In [13] this is done by introducing control inputs and a new operator $\odot$, which is an element-wise addition (addition as defined in normal algebra). This operator is not defined in the max-plus algebra, therefore the model cannot be considered as a purely max-plus algebraic model. In this section it is shown that changing the order of trains can be done entirely within the max-plus algebraic framework, without introducing a new operator.

Consider two trains running in the same direction over the same track again, as shown in Figure 3. Now if the order of the trains should be changed between train run l and i, equations (3) and (4) should be replaced by:

$$d_l(k) \geq d_i(k) \otimes \tau_{li}^{\text{headway}}(k) \quad (14)$$

$$a_l(k) \geq a_i(k) \otimes \tau_{li}^{\text{headway}}(k) \quad (15)$$

Replacing these constraints can be done by introducing a control variable $u_{il}(k) \in \{\varepsilon, 0\}$ and its negated control variable $\overline{u_{il}}(k) \in \{\varepsilon, 0\}/\{u_{il}(k)\}$. By multiplying equations (3) and (4) by $u_{il}(k)$ and multiplying equations (14) and (15) by $\overline{u_{il}}(k)$ we obtain:

$$d_i(k) \geq d_l(k) \otimes \tau_{il}^{\text{headway}}(k) \otimes u_{il}(k) \tag{16}$$

$$a_i(k) \geq a_l(k) \otimes \tau_{il}^{\text{headway}}(k) \otimes u_{il}(k) \tag{17}$$

$$d_l(k) \geq d_i(k) \otimes \tau_{li}^{\text{headway}}(k) \otimes \overline{u_{il}}(k) \tag{18}$$

$$a_l(k) \geq a_i(k) \otimes \tau_{li}^{\text{headway}}(k) \otimes \overline{u_{il}}(k) \tag{19}$$

If $u_{il}(k) = 0$, then $\overline{u_{il}}(k) = \varepsilon$ and the constraints become:

$$\begin{aligned}
d_i(k) &\geq d_l(k) \otimes \tau_{il}^{\text{headway}}(k) \otimes 0 = d_l(k) \otimes \tau_{il}^{\text{headway}}(k) \\
a_i(k) &\geq a_l(k) \otimes \tau_{il}^{\text{headway}}(k) \otimes 0 = a_l(k) \otimes \tau_{il}^{\text{headway}}(k) \\
d_l(k) &\geq d_i(k) \otimes \tau_{li}^{\text{headway}}(k) \otimes \varepsilon = \varepsilon \\
a_l(k) &\geq a_i(k) \otimes \tau_{li}^{\text{headway}}(k) \otimes \varepsilon = \varepsilon.
\end{aligned}$$

The first two equations are the same as (3) and (4) and the last two equations are always satisfied, since every element of the set $\mathbb{R}_{\max}$ is larger or equal than ε. As a result the order does not change from the default.

If $u_{il}(k) = \varepsilon$, then $\overline{u_{il}}(k) = 0$ and the constraints become:

$$\begin{aligned}
d_i(k) &\geq d_l(k) \otimes \tau_{il}^{\text{headway}}(k) \otimes \varepsilon = \varepsilon \\
a_i(k) &\geq a_l(k) \otimes \tau_{il}^{\text{headway}}(k) \otimes \varepsilon = \varepsilon \\
d_l(k) &\geq d_i(k) \otimes \tau_{li}^{\text{headway}}(k) \otimes 0 = d_i(k) \otimes \tau_{li}^{\text{headway}}(k) \\
a_l(k) &\geq a_i(k) \otimes \tau_{li}^{\text{headway}}(k) \otimes 0 = a_i(k) \otimes \tau_{li}^{\text{headway}}(k)
\end{aligned}$$

In this case the first two equations are always satisfied and the last two equations are the same as (14) and (15) resulting in a change in the order of the trains.

The same can be done for the headway constraints of two trains running in opposite direction:

$$d_i(k) \geq a_m(k) \otimes \tau_{im}^{\text{wait}(k)} \otimes u_{il}(k) \tag{20}$$

$$d_m(k) \geq a_i(k) \otimes \tau_{mi}^{\text{wait}(k)} \otimes \overline{u_{il}}(k) \tag{21}$$

With these adjustments reordering the trains can be modeled at the points in the network where it is physically possible. The resulting model is a switching max-plus-linear (SMPL) model. It is called a switching max-plus-linear model, since it can switch between behaviors (train orders). The SMPL model can be described as:

$$x(k) = A_0(u(k), k) \otimes x(k) \oplus A_1(k) \otimes x(k-1) \oplus r(k), \tag{22}$$

where $u(k)$ is the set containing all control variables $u_{il}(k)$ and $\overline{u_{il}}(k)$, and the elements of $A_0(u(k), k)$ are max-plus-linear functions in the control variables. As an example, consider the element $[A_0(u(k), k)]_{il}$. This element relates $d_i(k)$ to $d_l(k)$ and is determined by equation (16). The value of this element is

$$[A_0(u(k), k)]_{il} = \tau_{il}^{\text{headway}}(k) \otimes u_{il}(k),$$

which is a function that is linear in max-plus algebra; hence we call it a max-plus-linear function.

3 From the implicit to the explicit model

The model introduced in equation (22) has a specific structure called the implicit form. For an equation in the implicit form the state vector $x(k)$ depends not only on the state vector of the previous cycle (and the timetable reference), but also on itself. A disadvantage of an implicit model is that in order to determine the event times $x(k)$ the model needs to be iterated several times. By rewriting the model into its explicit form $x(k)$ can be calculated in a single iteration, but this calculation may be more time consuming than a single iteration of the implicit model, since the explicit model can be more complex than the implicit model.

From this point on $A_0(u(k), k)$ and $A_1(k)$ will be written as A_0 and A_1, since the SMPL model will be the only model considered. The SMPL model can be rewritten into its explicit form by determining A_0^* [1]:

$$x(k) = A_0^* \otimes A_1 \otimes x(k-1) \oplus A_0^* \otimes r(k) \tag{23}$$

where

$$A_0^* = \bigoplus_{p=0}^{\infty} A_0^{\otimes p}, \tag{24}$$

with

$$A_0^{\otimes p} = \underbrace{A_0 \otimes A_0 \otimes \ldots \otimes A_0}_{p \text{ times}} \tag{25}$$

and $A_0^{\otimes 0} = E$, 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.

For any railway model as defined in section 2, the infinite sum in equation (24) can in fact be limited to the dimension of the matrix minus one. This property can be derived from the fact that all event times are positive. As a result any circuit in the graph of the A_0 will have a positive weight. A circuit in the graph of A_0 implies a relation between each of the events in circuit to itself of the form $x_i = x_i + a$, with a the weight of the circuit. This kind of equations has no solution in $\mathbb{R}_{\max}$ for positive a, and therefore no circuits should exist in the graph of A_0. In max-plus algebra the value of $[A_0^{\otimes p}]_{ij}$ is the maximum weight of all paths of length p from j to i in the graph of A_0. If no path of length p exists, this value is ε. Since no circuits should exist, the length of the longest path in the graph of A_0 can at most be the dimension of the matrix minus one. As a result any matrix power $A^{\otimes i}$, with i larger than or equal to the dimension of the matrix, is equal to $\mathcal{E}$, where $\mathcal{E}$ is the zero matrix in max-plus algebra, with all entries equal to ε. Just like in regular algebra addition with a zero matrix does not change the matrix. Therefore, the sum in equation (24) A_0^* can be limited to the dimension of the matrix minus 1:

$$A_0^* = \bigoplus_{p=0}^{2q-1} A_0^{\otimes p}. \tag{26}$$

In the case of the SMPL model combinations of control inputs that correspond to infeasible orders of trains, will result in circuits of positive weight in the graph of A_0. These combinations of control inputs have to be identified and removed during the calculation of A_0^*. These combinations of control inputs can be identified by finding the diagonal elements in the matrices $A_0^{\otimes i}(k)$ that have a positive weight, since these correspond to equations of the form $x_i = x_i + [A_0^{\otimes i}(k)]_{ii}$, with $A_0^{\otimes i}(k)]_{ii}$ being the diagonal element with positive weight. By identifying and removing the infeasible combinations of control inputs resulting in these diagonal elements, a feasible explicit switching max-plus-linear model is found:

$$x(k) = A_{\text{exp}} \otimes x(k-1) \oplus A_0^{*,\text{feas}} \otimes r(k), \tag{27}$$

where $A_{\text{exp}} = A_0^{*,\text{feas}} \otimes A_1$, with $A_0^{*,\text{feas}}$ being equal to A_0^*, with the exception that the infeasible combinations of control inputs are removed from the matrix.

4 Rescheduling problem

With the models of the railway traffic introduced in Section 2, the propagation of the delay through the network and the effects of changing the order of trains on the delay propagation can be determined. This is needed to solve the rescheduling problem. The rescheduling problem is the problem of finding the order of the trains that minimizes a measure of the delay in the network, such as the total delay, or the maximum delay.

The rescheduling problem can be defined as a mixed integer linear programming (MILP) problem. In general an MILP problem is defined as

$$\min c^\top y \tag{28}$$

$$\text{s.t. } Ay \leq b \tag{29}$$

$$A_{\text{eq}} y = b_{\text{eq}} \tag{30}$$

where y is the vector of variables, $c^\top y$ is the objective function that needs to be minimized, and A, A_{eq}, b, and b_{eq} define the constraints on the variables.

In the case of the rescheduling problem, the vector of variables y contains the arrival and departure times of all train runs ($x(k)$) and the control variables ($u(k)$). The objective function $c^\top y$ is chosen such that the total delay in the system is minimized. This can be done by simply minimizing the sum of the arrival and departure times since the constraints (6),(7) ensure that the trains cannot arrive or depart before their scheduled arrival and departure times. No penalty is set on the control inputs.

The constraints on the variables are determined by the SMPL model. For the implicit

SMPL model the constraints on $x_i(k)$ can be written as

$$x_i(k) = \bigoplus_{j} \Big([A_0]_{ij} \otimes x_j\,(k) \Big) \oplus \bigoplus_{j=1}^{2q} \Big([A_1]_{ij} \otimes x_j\,(k-1) \Big) \oplus r_i(k)$$

$$x_i(k) = \max \bigg(\max_{j} \Big([A_0]_{ij} + x_j(k) \Big), \max_{j} \Big([A_1]_{ij} + x_j(k-1) \Big), r_i(k) \bigg)$$

This can be split into three sets of constraints:

$$\begin{aligned} x_i(k) &\geq [A_0]_{ij} + x_j(k), && \text{for all } j \\ x_i(k) &\geq [A_1]_{ij} + x_j(k-1), && \text{for all } j \\ x_i(k) &\geq r_i(k) \end{aligned}$$

MILP solvers cannot deal with control inputs with values $\varepsilon = -\infty$. This problem can be easily dealt with by introducing new binary variables $v_i(k)$ that are defined as

$$v_i(k) = \begin{cases} 0 & \text{if } u_i(k) = \ \ 0 \\ 1 & \text{if } u_i(k) = \ \ \varepsilon \end{cases} \tag{31}$$

and by replacing $u_i(k)$ by $\beta v_i(k)$ and $\overline{u_i}(k)$ by $\beta(1 - v_i(k))$, where $\beta \ll 0$. In this way an MILP problem of the form (28),(29),(30) is obtained.

For the explicit model the constraints on $x_i(k)$ can be written in a similar manner:

$$x_i(k) = \bigoplus_{j} \Big([A_{\text{exp}}]_{ij} \otimes x_j\,(k-1) \Big) \oplus \bigoplus_{j} \Big([A_0^{*,\text{feas}}]_{ij} \otimes r_j(k) \Big)$$

$$x_i(k) = \max \bigg(\max_{j} \Big([A_{\text{exp}}]_{ij} + x_j\,(k-1) \Big), \max_{j} \Big([A_0^{*,\text{feas}}]_{ij} + r_j(k) \Big) \bigg),$$

This can be split up into two sets of constraints

$$\begin{aligned} x_i(k) &\geq [A_{\text{exp}}]_{ij} + x_j\,(k-1) && \text{for all } j \\ x_i(k) &\geq [A_0^{*,\text{feas}}]_{ij} + r_j(k) && \text{for all } j \end{aligned}$$

Then by replacing the control variables in the same way as was done for the implicit model an MILP problem of the form (28),(29),(30) is obtained.

5 Test case

In order to test the computational performance of the rescheduling problem using the implicit and explicit SMPL models a small railway traffic network is considered. The network is shown in Figure 5.

The model has a timetable period of 30 minutes; 11 trains are modeled to run over the network, resulting in 31 train runs, 62 continuous variables, and 25 control inputs. The rescheduling problem using the implicit SMPL model has 319 constraints, while the explicit SMPL model has 17067 constraints.

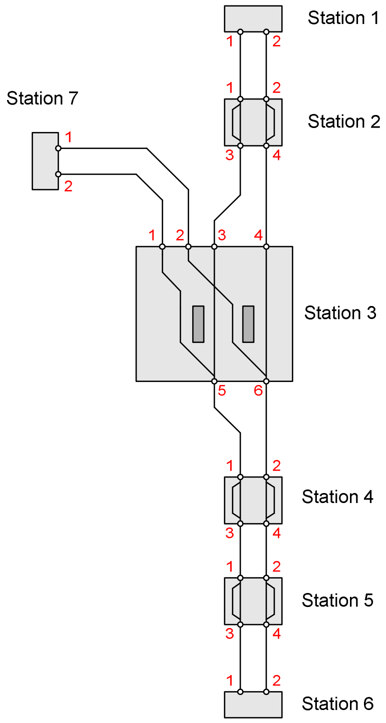

Figure 5: Small railway traffic network

To evaluate the computational performance a set of 50 scenarios is considered. In the scenarios 20% of the trains are delayed by a randomly selected delay based on a truncated Weibull distribution, with shape parameter of 0.8 and a scale parameter of 20. The delays are cut off at a maximum value of 40 minutes. The sum of delays for the 50 scenarios is shown in Figure 6. The black bars correspond to the uncontrolled network, while the white bars correspond to the optimal solution. The solutions of the implicit and explicit model are exactly the same. The computation time required to solve the 50 scenarios is shown in Figure 7 on a logarithmic scale. All results are also shown in Table 1.

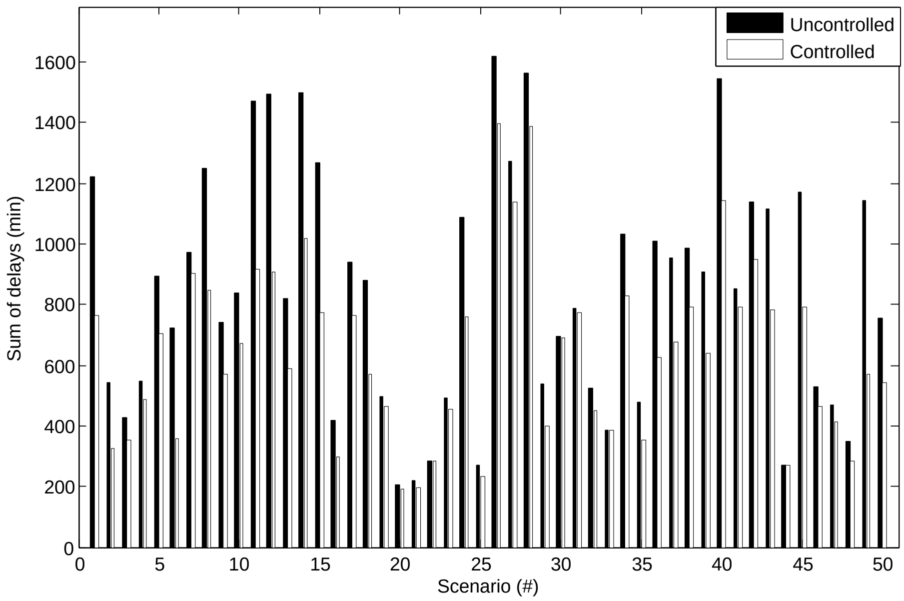

Figure 6: Sum of delays for the 50 scenarios for the uncontrolled (black bars) and optimally controlled (white bars) case.

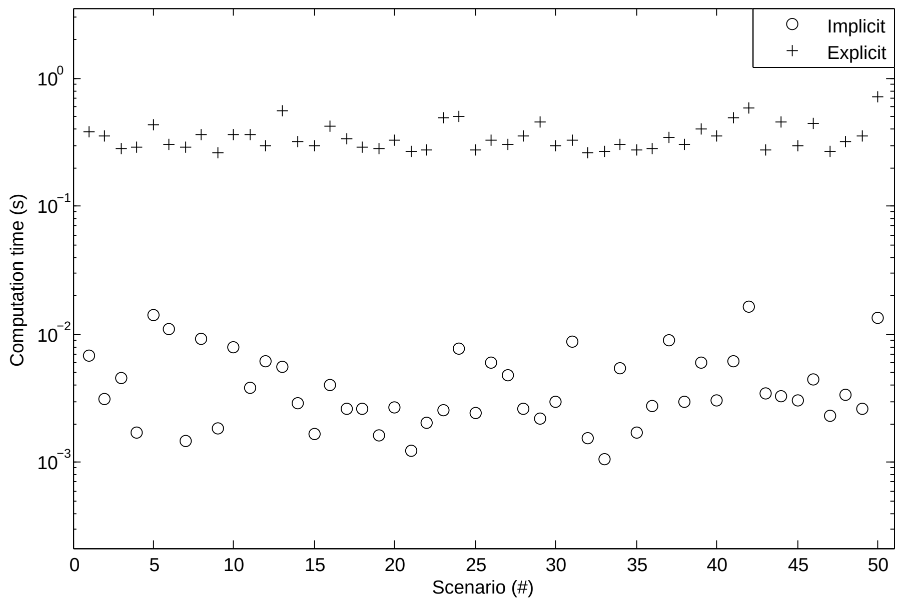

Figure 7: Computation time of the solver for the 50 scenarios for the implicit (+) and explicit (o) models.

Table 1: The results of the 50 scenarios of the test case for a time horizon of 30 minutes.

Scenario (#)	Delay Uncon. (min)	Delay Con. (min)	Comp. time Imp. (s)	Comp. time Exp. (s)
1	1221	766	0.0069	0.3836
2	545	325	0.0031	0.3550
3	425	355	0.0046	0.2863
4	549	485	0.0017	0.2931
5	894	706	0.0143	0.4384
6	722	356	0.0110	0.3058
7	971	901	0.0015	0.2865
8	1249	848	0.0092	0.3609
9	742	569	0.0018	0.2648
10	840	674	0.0079	0.3601
11	1472	915	0.0038	0.3655
12	1493	906	0.0062	0.2963
13	819	587	0.0056	0.5620
14	1498	1018	0.0029	0.3186
15	1268	773	0.0017	0.3008
16	419	299	0.0040	0.4241
17	939	762	0.0026	0.3407
18	881	571	0.0026	0.2865
19	496	464	0.0016	0.2827
20	203	191	0.0027	0.3284
21	218	198	0.0012	0.2698
22	282	282	0.0020	0.2791
23	494	457	0.0025	0.4980
24	1087	758	0.0077	0.5105
25	270	235	0.0024	0.2784
26	1621	1398	0.0060	0.3303
27	1271	1140	0.0048	0.3081
28	1562	1390	0.0026	0.3522
29	537	401	0.0022	0.4592
30	697	689	0.0030	0.2981
31	788	772	0.0088	0.3317
32	525	452	0.0015	0.2637
33	388	388	0.0011	0.2712
34	1034	830	0.055	0.3031
35	477	355	0.0017	0.2777
36	1009	626	0.0027	0.2806
37	956	679	0.0090	0.3439
38	984	793	0.0030	0.3019
39	907	638	0.0061	0.4060
40	1544	1144	0.0031	0.3578
41	851	791	0.0061	0.4928

42	1138	948	0.0166	0.5934
43	1117	784	0.0034	0.2750
44	269	269	0.0033	0.4512
45	1170	795	0.0030	0.3002
46	530	465	0.0044	0.4486
47	468	414	0.0023	0.2672
48	351	286	0.0034	0.3197
49	1141	569	0.0026	0.3529
50	755	541	0.0134	0.7097

From the results it is clear that solving the rescheduling problem using the explicit model takes a lot more time than solving it using the implicit model. At best it is about 28 times as slow and in the worst case it is 254 times as slow.

6 Discussion

We have proposed an adaptation of the SMPL model for railway networks introduced in [13]. Our approach allows the model to be described completely within the max-plus algebra. Next, it has been shown how the implicit SMPL model can be converted into its explicit form. For a case study of a small railway network we have compared the time needed to solve the rescheduling problem, using either of these models. From these results the conclusion can be drawn that solving the rescheduling problem using the explicit model requires more time than solving the problem using the implicit model. So currently there is no benefit in using the explicit model instead of the implicit model. Therefore, the next step in our research will to reduce the computation time needed to solve the rescheduling problem for both models. One way of doing this is by limiting the control freedom. The control freedom can be limited by removing some (combinations of) control inputs that have little or no effect on the railway network. This will result in smaller MILP problems. It is expected that the explicit model will benefit more from these methods to limit the control freedom because in the implicit model no combinations of control inputs can be removed, since the constraints resulting from combinations of control inputs are not explicitly modeled but are modeled through the iterative structure of the model. For the implicit model these combinations of control inputs can only be excluded by adding a constraint that is violated if this combination is chosen. On the other hand, in the explicit model these combinations of control inputs are explicitly modeled and can therefore be easily removed, resulting in a larger reduction in the number of constraints for the MILP problem based on this model.

Acknowledgments

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs.

References

[1] F. Baccelli, G. Cohen, G. Oslder, and J. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Wiley, 1992.

[2] J.G. Braker. Max-algebra modelling and analysis of time-dependent transportation networks. In *Proceedings of the 1st European Control Conference*, pages 1831–1836, Grenoble, France, July 1991.

[3] J.G. Braker. *Algorithms and Applications in Timed Discrete Event Systems*. Delft University of Technology, Delft, The Netherlands, 1994.

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

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

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

[7] Rob M.P. Goverde. Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B: Methodological*, 41(2):179 – 201, 2007.

[8] Rob M.P. Goverde. A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(3):269 – 287, 2010.

[9] P. Kecman, F. Corman, A. D'Ariano, and R. Goverde. Rescheduling models for network-wide railway traffic management. In *Proceedings of the 12th Conference on Advanced Systems for Public Transport (CASPT 12)*, pages 1–30, Santiago, Chile, July 2012.

[10] J. Törnquist Krasemann. 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, 2012.

[11] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004.

[12] 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*, pages 9537–9542, Milan, Italy, August–September 2011.

[13] 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*, pages 7366–7371, Maui, Hawaii, December 2012.