

Technical report 11-021

# **A permutation-based algorithm to optimally reschedule trains in a railway traffic network\***

T.J.J. van den Boom, N. Weiss, W. Leune, R.M.P. Goverde, and  
B. De Schutter

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

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,” *Proceedings of the 18th IFAC World Congress*, Milan, Italy, pp. 9537–9542, Aug.–Sept. 2011. doi:[10.3182/20110828-6-IT-1002.02352](https://doi.org/10.3182/20110828-6-IT-1002.02352)

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.bartdeschutter.org/abs/11\\_021](https://pub.bartdeschutter.org/abs/11_021)

# A permutation-based algorithm to optimally reschedule trains in a railway traffic network

Ton J.J. van den Boom\* Nicolás Weiss\* Wilko Leune\*  
Rob M.P. Goverde\*\* Bart De Schutter\*

\* *Delft University of Technology, Delft Center for Systems and Control, Delft, The Netherlands (e-mail: {a.j.j.vandenboom, n.m.weiss, b.deschutter}@tudelft.nl).*

\*\* *Delft University of Technology, Department of Transport and Planning, Delft, The Netherlands (e-mail: r.m.p.goverde@tudelft.nl).*

---

**Abstract:** In this paper we discuss dynamic traffic management of railway traffic networks at an operational level. We design a model predictive controller based on measurements of the actual train positions. The core of the model predictive control approach is the railway traffic model, for which a switching max-plus linear system is proposed. If the model is affine in the controls, the optimisation problem can be recast as a mixed-integer linear programming problem. To this end we present a permutation-based algorithm to model the rescheduling of trains running on the same track. We apply the algorithm to a simple railway traffic network simulation model and show a significant reduction of delays compared to the uncontrolled case.

*Keywords:* Switching max-plus linear models, model predictive control, mixed-integer linear programming, railway traffic management.

---

## 1. INTRODUCTION

Current practice in the operational-level management of railway traffic networks is mostly based on predefined rules and on the ability of traffic controllers and train dispatchers to detect and avoid conflicting situations. Delays caused by technical failures, fluctuation of passenger volumes, and/or weather conditions can be partly absorbed by a stable and robust timetable (Goverde, 2007). In the case of large delays, network managers might be forced to re-route or to change the order of trains, break connections, or even cancel a scheduled service to prevent the accumulation of delays in the network. In this paper we design a predictive feedback controller that computes the most effective actions, based on measurements of the actual train positions. The control measures are restricted to changing the order of trains running on the same track.

A railway network with rigid connection constraints and a fixed routing schedule can be modelled using max-plus-linear (MPL) models (Heidergott and de Vries, 2001). An MPL model is linear in the max-plus algebra (Baccelli et al., 1992), which has maximisation and addition as its basic operations. Max-plus-linear systems can be characterised as discrete event systems in which only synchronisation and no concurrency or choice occurs (Baccelli et al., 1992). In the railway context, synchronisation means that some trains should give predefined connections to other trains, and a fixed routing schedule means that the order of departure is fixed. In this paper we model a controlled railway system using the *switching* max-plus-linear system

description of van den Boom and De Schutter (2006). In this description we use a number of MPL models, each model corresponding to a specific mode, describing the network by a different set of connection and order constraints. We control the system by switching between different modes, allowing us to change the order of trains to minimise the delays of all trains in the network while considering the cost of the control actions.

In the case that only changes in the order of subsequent trains are allowed, we have shown in van den Boom and De Schutter (2007) how the resulting optimisation problem can be recast as a mixed-integer linear programming (MILP) problem. In this paper we present an algorithm to extend the model and the approach to change the order of *any* number of trains and we show that the problem still yields an MILP. We show that for a simple network model the proposed method yields good results.

## 2. MODEL

Consider a periodic railway operations system that follows a schedule with period  $T$ . In nominal operation mode, we assume that all the trains follow a pre-scheduled route, with a fixed train order and predefined connections. If for any of the reasons mentioned before delays are introduced in the network, it might be advantageous to change the train order so as to minimise delays. In this case we will operate in a perturbed mode with an associated new schedule. First we discuss the nominal operation mode.

---

\* Research funded by the Dutch Technology Foundation STW project ‘Model-Predictive Railway Traffic Management’ (11025).

## 2.1 Nominal operation

Consider an unperturbed railway operations system where each train running on each track of the railway network has a number assigned to it. For the sake of simplicity we will say ‘train  $i$ ’ to denote the (physical) train on (virtual) track  $i$ , and ‘station  $i$ ’ to denote the (virtual) station at the beginning of track  $i$  (cf. Figure 1). Let  $n$  be the number of ‘virtual’ tracks in the network. We say virtual to denote that some of the virtual tracks or stations may actually be the same physical track or station (corresponding to different trains using the same track or station). This means that the actual number of tracks is usually smaller than  $n$ .

We assume that overtaking actions can only take place at stations and that sufficient infrastructure capacity is available.

Let  $d_i(k)$  be the time instant at which train  $i$  departs from its departure station for the  $k$ th time, and let  $a_i(k)$  be the time instant at which train  $i$  arrives at its arrival station for the  $k$ th time. Let  $r_i(k)$  ( $r_{i+n}(k)$ ) be the scheduled departure (arrival) time for this train according to the timetable.

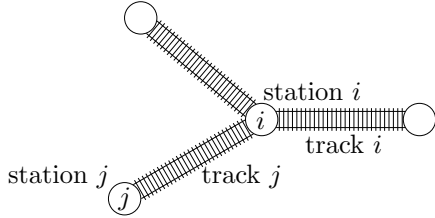


Fig. 1. A part of a railway network.

Let  $p_i$  be the predecessor track of train  $i$ , i.e., the track that ended at station  $i$ , and let  $C_i(k)$  be the set of trains that give a connection to train  $i$  in the  $k$ th cycle. Let  $\mathcal{F}_i(k)$  be the set of trains that move over the same track as train  $i$ , in the same direction as train  $i$ , and that are scheduled before train  $i$  in the  $k$ th cycle. Let  $\mathcal{W}_i(k)$  be the set of trains that move over the same track as train  $i$ , in the opposite direction of train  $i$ , and are scheduled before train  $i$  in the  $k$ th cycle. Furthermore, let  $t_i(k)$  be the running time of train  $i$  in the  $k$ th cycle. Define a minimum connection time  $c_{ij}(k)$  for passengers to get from train  $j$  to train  $i$  for each train  $j \in C_i(k)$  in the  $k$ th cycle, and define a minimum dwell time  $s_j(k)$  of train  $j$  at station  $j$  in the  $k$ th cycle to allow passengers to board or alight the train. Finally, define a minimum headway time  $h_{ij}(k)$  between two different trains  $i$  and  $j$  moving over the same track and in the same direction in the  $k$ th cycle, and a minimum headway time  $w_{ij}(k)$  between two different trains moving over the same track and in the opposite direction in the  $k$ th cycle. Throughout this paper the minimum headway times for arrival and departure constraints are assumed to be equal.

The departures and arrivals of train  $i$  are subject to the following constraints:

- *Time schedule constraint:*

$$d_i(k) \geq r_i(k), \quad (1)$$

$$a_i(k) \geq r_{i+n}(k), \quad (2)$$

where  $r_i(k) = r_i(0) + kT$  with  $r_i(0)$  the initial scheduled departure time, and with a similar definition for the arrivals.

- *Running time constraint:*

$$a_i(k) \geq d_i(k - \delta_{ii}) + t_i(k), \quad (3)$$

where  $\delta_{ii} = 0$  if train  $i$  is scheduled to arrive at its destination in the same cycle as its departure, and  $\delta_{ii} = m$  if it arrives  $m$  cycles after its departure.

- *Continuity constraints:*

$$d_i(k) \geq a_{p_i}(k - \delta_{ip_i}) + s_{p_i}(k), \quad (4)$$

where  $\delta_{ip_i} = m$  if the  $(k - m)$ th train  $p_i$  arriving at the physical station corresponding to virtual station  $i$  continues as the  $k$ th train  $i$ .

- *Connection constraints:*

$$d_i(k) \geq a_j(k - \delta_{ij}) + c_{ij}(k), \quad \forall j \in C_i(k), \quad (5)$$

where  $\delta_{ij} = m$  if the  $(k - m)$ th train  $j$  gives a connection to the  $k$ th train  $i$ .

- *Headway constraints:*

$$d_i(k) \geq d_j(k - \delta_{ij}) + h_{ij}(k), \quad \forall j \in \mathcal{F}_i(k), \quad (6)$$

$$a_i(k) \geq a_j(k - \delta_{ij}) + h_{ij}(k), \quad \forall j \in \mathcal{F}_i(k), \quad (7)$$

where  $\delta_{ij}$  is defined similarly as above.

- *Meeting constraints:*

$$d_i(k) \geq a_j(k - \delta_{ij}) + w_{ij}(k), \quad \forall j \in \mathcal{W}_i(k), \quad (8)$$

where again  $\delta_{ij}$  is defined similarly as above.

Note that in nominal operation generally all  $\delta_{ij}$ 's are equal to zero or one, but in perturbed operation other values of  $\delta_{ij}$  are possible. Moreover, note that in the general case the  $\delta_{ij}$ 's could depend on  $k$ .

Since a train is allowed to depart as soon as all constraints are satisfied, we have

$$d_i(k) = \max \left\{ r_i(k), a_{p_i}(k - \delta_{ip_i}) + s_{p_i}(k), \max_{j \in C_i(k)} (a_j(k - \delta_{ij}) + c_{ij}(k)), \max_{j \in \mathcal{F}_i(k)} (d_j(k - \delta_{ij}) + h_{ij}(k)), \max_{j \in \mathcal{W}_i(k)} (a_j(k - \delta_{ij}) + w_{ij}(k)) \right\}, \quad (9a)$$

$$a_i(k) = \max \left\{ r_{i+n}(k), d_i(k - \delta_{ii}) + t_i(k), \max_{j \in \mathcal{F}_i(k)} (a_j(k - \delta_{ij}) + h_{ij}(k)) \right\}. \quad (9b)$$

Note that in an undisturbed, well-defined timetable the term  $r_i(k)$  in (9a) will be the largest. However, if due to unforeseen circumstances one of the trains has a delay, the corresponding term can become larger than the others, and train  $i$  will depart later than the scheduled departure time  $r_i(k)$ .

Consider a network with  $n$  trains and define the vectors  $x(k) = [d_1(k), \dots, d_n(k), a_1(k), \dots, a_n(k)]^T \in \mathbb{R}^{2n}$  and  $r(k) \in \mathbb{R}^{2n}$ . By defining  $\varepsilon = -\infty$ ,  $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$ , and appropriate matrices  $A_m \in \mathbb{R}_\varepsilon^{2n \times 2n}$ ,  $m = m_1, \dots, m_2$  with  $m_1 = \min_{i,j}(\delta_{ij})$  and  $m_2 = \max_{i,j}(\delta_{ij})$ , we can rewrite (9a) and (9b) as:

$$x_i(k) = \max \left( r_i(k), \max_{j,m} \left( x_j(k - m) + [A_m]_{ij} \right) \right), \quad (10)$$

where  $[A_m]_{ij}$  is the  $(i, j)$ th entry of  $A_m$ .

Now we introduce the notation of the max-plus algebra (Baccelli et al., 1992). The max-plus-algebraic addition ( $\oplus$ ) and multiplication ( $\otimes$ ) are defined as:

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

for  $x, y \in \mathbb{R}_\varepsilon$ , and

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

for  $A, B \in \mathbb{R}_\varepsilon^{m \times n}$ ,  $C \in \mathbb{R}_\varepsilon^{n \times p}$ . In regards to the order of evaluation  $\otimes$  has preference over  $\oplus$ .

The max-plus-algebraic zero matrix  $\mathcal{E}$  is defined as:

$$\mathcal{E}_{ij} = \varepsilon, \quad \forall i, j.$$

In max-plus notation (10) becomes

$$x_i(k) = r_i(k) \oplus \bigoplus_{j=1}^{2n} \bigoplus_{m=m_1}^{m_2} x_j(k-m) \otimes [A_m]_{ij},$$

and in matrix-notation we obtain

$$x(k) = \bigoplus_{m=m_1}^{m_2} A_m \otimes x(k-m) \oplus r(k). \quad (11)$$

## 2.2 Perturbed operation

In nominal operation we have assumed that some trains should give predefined connections to other trains, and that the order of trains on the same track is fixed. However, if one of the preceding trains has a too large delay, then it is sometimes better — from a global performance viewpoint — to let a connecting train depart anyway or to change the departure order on a specific track. This is done in order to prevent an accumulation of delays in the network. In this paper we consider the switching between different operation modes, where each mode corresponds to a specific order of train departures and arrivals. Note that any change of train order leads to a new model, similar to the nominal equation (11), but now with adapted system matrices  $A_m(\ell(k))$  for the  $\ell$ th mode. A system operating in a perturbed mode  $\ell(k)$  can be described as

$$x(k) = \bigoplus_{m=m_1}^{m_2} A_m(\ell(k)) \otimes x(k-m) \oplus r(k), \quad (12)$$

where mode  $\ell(k) = 0$  corresponds to the nominal timetable.

## 3. THE RAILWAY CONTROL PROBLEM

### 3.1 Timing aspects

Discrete event systems are different from conventional time-driven systems in the sense that the event counter  $k$  is not directly related to a specific time. Let  $t$  be a given time instant and let  $k$  be such that  $x(k-m_2), \dots, x(k-1)$  are completely known, i.e.,  $x_i(k-j) \leq t \quad \forall i, \forall j \in \{1, 2, \dots, m_2\}$ . So at time instant  $t$  in cycle  $k$ , some of the components of  $x(k)$  may already be known while others may still lie in the future. In (van den Boom and De

Schutter, 2007) we have presented a method to address the timing issues in the control of switching MPL systems. This method is made more specific here for the model in (9a) and (9b).

The case of full state information is considered, since the components of  $x(k)$  correspond to departure and arrival times, which are in general easy to measure.

Furthermore, we may have measurements of trains in cycle  $k$  that have already departed or arrived at time instant  $t$ . If so, we denote these as  $d_{\text{past}}(k|t)$  and  $a_{\text{past}}(k|t)$  respectively. Sometimes there is information available about the estimated running time for trains that have not yet arrived at their destination at time  $t$ . With this information we can make an estimation  $\hat{t}_{\text{est}}(k|t)$  of the future running times. If no further information is available on a specific running time we take the nominal running time  $[\hat{t}_{\text{est}}(k|t)]_i = t_i(k)$ . All these values can be substituted in the matrices  $A_m(\ell(k))$  to make the future system description  $\hat{A}_m(\ell(k|t))$  as accurate as possible.

Due to the fact that time does not explicitly enter the max-plus recurrence in (12) it could happen that if an event is delayed by a precedence constraint and if due to a rescheduling action this hindering constraint is removed, then the delayed event could, in theory, be rescheduled in the past. To avoid this issue it should be considered that a departure event cannot be rescheduled before the current decision time  $t$ .

### 3.2 Control problem

Let  $N_p$  be the prediction horizon and define the set  $\mathcal{U}(k+j|t) \subset \{0, 1\}^{n_u}$  for  $j = 0, 1, \dots, N_p$  with  $n_u$  the dimension of the control space, as the set of possible future control actions for the  $k+j$ th cycle at time instant  $t$  with the relation between  $k$  and  $t$  as defined in Section 3.1. Note that this includes the fact that certain control actions may no longer be feasible, as at time instant  $t$  some trains might have already departed and therefore their order can no longer be changed. So, a mode  $\ell(k)$  will be encoded as a binary array with  $n_u$  bits (see Section 4 for details).

To select the optimal set of possible future control actions, we define the following optimal control problem at time instant  $t$ :

$$\begin{cases} \min_{u(k|t), u(k+1|t), \dots, u(k+N_p|t)} J(k|t) \\ \hat{x}(k+j|t) = \bigoplus_{m=m_1}^{m_2} \hat{A}_m(\ell(k+j|t)) \otimes \hat{x}(k+j-m|t) \oplus r(k+j) \\ \hat{x}(k-i) = x(k-i) \text{ for } i = 1, \dots, m_2 \\ u(k+j|t) \in \mathcal{U}(k+j|t) \end{cases} \quad (13)$$

where the performance index  $J(k|t)$  is given by

$$J(k|t) = \sum_{j=0}^{N_p} \left( \sum_{i=1}^{2n} \sigma_i \hat{e}_i(k+j|t) + \sum_{l=1}^{n_u} \rho_l u_l(k+j|t) \right). \quad (14)$$

Here  $\hat{e}(k+j|t)$  is the vector with the expected delays ( $\hat{e}_i(k+j|t) = \hat{x}_i(k+j|t) - r_i(k+j) \geq 0$ ), and  $\sigma_i, \rho_l$  are positive weighting scalars. The first term of (14) is related to the sum of all predicted delays, and the second term

denotes the penalty for all switched train orders during cycle  $k + j$ .

To compute the predictions of  $\hat{x}(k + j|t)$  in (13) we made use of the fact that at time  $t$  we have  $d_{\text{past}}(k|t)$ ,  $a_{\text{past}}(k|t)$ , and  $\hat{t}_{\text{est}}(k + j|t)$  available, so that the estimates  $\hat{A}_m(\ell(k + j|t))$  of all future  $A_m(\ell(k + j))$  could be computed.

In principle we now have all elements to solve the optimal control problem (13). Note that if the railway timetable is well-defined and there is some margin in the schedule, there will always be an integer  $N$  such that for  $N_p \geq N$  in the nominal case ( $u(k + j|t) = 0 \forall j \geq 0$ ) the delays will vanish ( $\hat{e}(k + j|t) = 0 \forall j \geq N_p$ ) (Heidergott and de Vries, 2001). By choosing  $N_p = N$  in (14) we are sure that enough delay terms are taken into account. In many cases a smaller value for  $N_p$  will be sufficient. A major advantage of a small prediction horizon  $N_p$  is that the computational complexity of the optimisation problem is drastically reduced.

#### 4. MODE ENCODING USING AN AFFINE MODEL

In (van den Boom and De Schutter, 2007) we have shown that by restricting the change in departure order of two *successive* trains, the railway network model can be written in affine form with respect to the controls, and the optimisation problem in (13) can be recast into an MILP problem. Affinity with respect of the controls means that the system matrix can be written as:

$$\hat{A}_m(\ell(k)) = \hat{A}_{m,0} + \sum_{v=1}^{n_u} \hat{A}_{m,v}(k) u_v(k), \quad (15)$$

with  $u_v(k) \in \{0, 1\}$  and  $\hat{A}_{m,v}(k) \in \mathbb{R}_{\varepsilon}^{2n \times 2n}$ ,  $v = 1, 2, \dots, n_u$ . Note that matrix  $\hat{A}_m(\ell(k))$  depends now on the control  $u(k)$  and a mode  $\ell(k)$  is determined by a certain combination of control entries  $u_v(k)$ , i.e.,  $\ell$  depends on  $u(k)$ :  $\ell(k) = L(u(k))$ . Equation (15) describes the actual configuration of the network  $\hat{A}_m(\ell(k))$  as the sum of the nominal network configuration  $\hat{A}_{m,0}$  and so called mode matrices  $\hat{A}_{m,v}(k)$ . To allow for the required manipulations stated by (15) we substitute all elements of  $\hat{A}_{m,0}$  that are equal to  $\varepsilon$  by  $\beta$ , a large negative real number ( $\beta \ll 0$ ).

In this section we extend the model to the case of changing the order of *non-successive* trains scheduled to run on the same physical track based on a permutation method. In Section 5 we show how the resulting optimisation problem in (13) can be recast as an MILP problem.

##### 4.1 Permutation method

Let us assume that  $\mathcal{T} = \{1, 2, \dots, n_t\}$  is the set containing all physical tracks in the network with  $n_t$  the total number of physical tracks. The method computes for each physical track all the allowed permutations of the currently scheduled train sequence. A permutation is called allowed if it does not violate a continuity constraint as introduced in Section 2.1.

In the following a superscript  $\tau$  denotes the association to the  $\tau$ th physical track with  $\tau \in \mathcal{T}$ . Each permutation of trains on each physical track  $\tau \in \mathcal{T}$  is described with a binary control vector

$$u^{(\tau)}(k) = [u_1^{(\tau)}(k), \dots, u_{\sigma_{\tau}(k)}^{(\tau)}(k)]^T,$$

with  $u_v^{(\tau)}(k) \in \{0, 1\}$ ,  $v = 1, 2, \dots, \sigma_{\tau}(k)$  and

$$\sigma_{\tau}(k) = \binom{n_{\tau}(k)}{2} = \frac{n_{\tau}(k)(n_{\tau}(k) - 1)}{2}$$

the number of control entries for the  $\tau$ th physical track and  $n_{\tau}(k)$  the number of trains scheduled to run on the  $\tau$ th physical track in cycle  $k$ . Note that  $\sigma_{\tau}(k)$  represents the number of 2-permutations of  $n_{\tau}(k)$  trains on the  $\tau$ th physical track in cycle  $k$ . Therefore, a control  $u_v^{(\tau)}(k)$ ,  $v = 1, 2, \dots, \sigma_{\tau}(k)$  is then to be associated to the permutation of the order of two specific trains scheduled to run on the same physical track  $\tau$  with respect to the currently scheduled sequence of trains. This means that if in a given permutation the order of two trains is swapped, then the associated control will be equal to one. On the other hand, if the order of the two trains does not change with respect to the currently scheduled sequence, then the associated control will be equal to zero.

To define the mode matrices  $\hat{A}_{m,v}(k)$  in (15) two cases have to be accounted for: the case that a control is associated to changing the order of two trains that are scheduled in the same direction, and the case that a control is associated to changing the order of two trains that are scheduled in opposite directions. In either case the mode matrices will implement the removal of an existing precedence constraint between two trains and the addition of a new one. Let us assume that a control  $u_v^{(\tau)}(k)$  is associated to the allowed permutation of the  $p$ th and  $q$ th train on track  $\tau$  in cycle  $k$ , with  $q < p$ , i.e., the  $q$ th train is originally scheduled before the  $p$ th train in a cycle. Then the corresponding mode matrix  $\hat{A}_{m,v}^{(\tau)}(k)$  can be written as

$$[\hat{A}_{m,v}^{(\tau)}(k)]_{ij} = \begin{cases} \beta - h_{pq}(k) & \text{if } (i, j) = (p, q), \\ h_{qp}(k) - \beta & \text{if } (i, j) = (q, p), \\ \beta - h_{pq}(k) & \text{if } (i, j) = (n + p, n + q), \\ h_{qp}(k) - \beta & \text{if } (i, j) = (n + q, n + p), \\ 0 & \text{otherwise.} \end{cases}$$

In the case that train  $q$  is scheduled before and in opposite direction as train  $p$ , then matrix  $\hat{A}_{m,v}^{(\tau)}(k)$  is written as

$$[\hat{A}_{m,v}^{(\tau)}(k)]_{ij} = \begin{cases} \beta - w_{pq}(k) & \text{if } (i, j) = (p, n + q), \\ w_{qp}(k) - \beta & \text{if } (i, j) = (q, n + p), \\ 0 & \text{otherwise.} \end{cases}$$

Now (15) can be written as

$$\hat{A}_m(\ell(k)) = \hat{A}_{m,0} + \sum_{\tau=1}^{n_t} \sum_{v=1}^{\sigma_{\tau}(k)} \hat{A}_{m,v}^{(\tau)}(k) u_v^{(\tau)}(k), \quad (16)$$

where affinity with respect of the controls is still preserved. The total number of controls  $n_u$  is then calculated as

$$n_u = \sum_{\tau=1}^{n_t} \sigma_{\tau}(k) = \sum_{\tau=1}^{n_t} \frac{n_{\tau}(k)(n_{\tau}(k) - 1)}{2},$$

where we have assumed that the total number of trains in the network is constant over all cycles  $k$ .

#### 5. REFORMULATION AS A MIXED-INTEGER LINEAR PROGRAMMING PROBLEM

Now we show that the model predictive control problem (13) with  $\hat{A}_m(\ell(k))$  given by (16) can be recast into an

MILP problem. Assuming that in general  $m_1 = 0$  and  $m_2 \geq m_1$ , we outline now the main ideas behind this transformation. For the sake of simplicity of notation we drop the notation  $\hat{\cdot}$  and  $|t$  for a prediction from now on. Define the vectors

$$\begin{aligned}\tilde{x}(k) &= [x^T(k), \dots, x^T(k + N_p)]^T, \\ \tilde{u}(k) &= [u^T(k), \dots, u^T(k + N_p)]^T, \\ \tilde{z}(k) &= [x^T(k - 1), \dots, x^T(k - m_2)]^T, \\ \tilde{\ell}(k) &= [\ell(k), \dots, \ell(k + N_p)]^T, \\ \tilde{r}(k) &= [r^T(k), \dots, r^T(k + N_p)]^T,\end{aligned}$$

where  $\tilde{x}(k)$  represents the partially known or completely unknown states and  $\tilde{z}(k)$  represents the completely known states at cycle  $k$  as introduced in Section 3.1. Note that by defining the matrices

$$\tilde{A}(\tilde{\ell}(k)) = \begin{bmatrix} A_0(\ell(k)) & A_{-1}(\ell(k)) & \cdots & A_{-N_p}(\ell(k)) \\ A_1(\ell(k+1)) & A_0(\ell(k+1)) & \cdots & A_{1-N_p}(\ell(k+1)) \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_p}(\ell(k+N_p)) & A_{N_p-1}(\ell(k+N_p)) & \cdots & A_0(\ell(k+N_p)) \end{bmatrix}$$

where  $A_m(k+j) = \mathcal{E}$  for  $m < m_1$  and for  $m > m_2$ , and  $\tilde{B}(\tilde{\ell}(k)) =$

$$\begin{bmatrix} A_1(\ell(k)) & \cdots & A_{m_2-1}(\ell(k)) & A_{m_2}(\ell(k)) \\ A_2(\ell(k+1)) & \cdots & A_{m_2}(\ell(k+1)) & \mathcal{E} \\ \vdots & \ddots & \vdots & \vdots \\ A_{m_2}(\ell(k+m_2-1)) & \cdots & \mathcal{E} & \mathcal{E} \\ \mathcal{E} & \cdots & \mathcal{E} & \mathcal{E} \\ \vdots & \ddots & \vdots & \vdots \\ \mathcal{E} & \cdots & \mathcal{E} & \mathcal{E} \end{bmatrix},$$

we can write

$$\tilde{x}(k) = \tilde{A}(\tilde{\ell}(k)) \otimes \tilde{x}(k) \oplus \tilde{B}(\tilde{\ell}(k)) \otimes \tilde{z}(k) \oplus \tilde{r}(k). \quad (17)$$

Note that  $\tilde{x}(k)$  appears in both sides of (17). This is not a major problem if the matrix  $\tilde{A}(\tilde{\ell}(k))$  has a strictly lower triangular structure, which can always be achieved by a renumbering of the departures and arrivals (Goverde, 2010). Furthermore, due to (16) the matrices  $\tilde{A}(\tilde{\ell}(k))$  and  $\tilde{B}(\tilde{\ell}(k))$  will be affine in  $\tilde{u}(k)$ , and there exist matrices  $\tilde{A}_v$  and  $\tilde{B}_v$  such that:

$$\begin{aligned}\tilde{A}(\tilde{\ell}(k)) &= \tilde{A}_0 + \sum_{v=1}^{n_u} \tilde{A}_v \tilde{u}_v(k), \\ \tilde{B}(\tilde{\ell}(k)) &= \tilde{B}_0 + \sum_{v=1}^{n_u} \tilde{B}_v \tilde{u}_v(k).\end{aligned}$$

Note that  $\tilde{\ell}(k)$  is a function of  $\tilde{u}(k)$ , which can be expressed as  $\tilde{\ell}(k) = \tilde{L}(\tilde{u}(k))$ . The objective function  $J(k)$  is linear in  $\tilde{u}(k)$  and  $\tilde{x}(k)$ , and can be written as:

$$J(k) = c_e^T \tilde{x}(k) + c_u^T \tilde{u}(k), \quad (18)$$

where the constant term  $-c_e^T \tilde{r}(k)$  has been omitted since it does not affect the optimisation. Let us now show that (17) can be written as

$$\begin{aligned}\tilde{x}_i(k) &= \max(\tilde{r}_i(k), \max_j(\tilde{x}_j(k) + [\tilde{A}(\tilde{\ell}(k))]_{ij})), \\ &\quad \max_j(\tilde{z}_j(k) + [\tilde{B}(\tilde{\ell}(k))]_{ij})),\end{aligned} \quad (19)$$

which can be transformed into

$$\begin{cases} \tilde{x}_i(k) \geq \tilde{r}_i(k), \\ \tilde{x}_i(k) \geq \tilde{x}_j(k) + [\tilde{A}_0]_{ij} + \sum_{v=1}^{n_u} [\tilde{A}_v]_{ij} \tilde{u}_v(k) \quad \forall j, \\ \tilde{x}_i(k) \geq \tilde{z}_l(k) + [\tilde{B}_0]_{il} + \sum_{v=1}^{n_u} [\tilde{B}_v]_{il} \tilde{u}_v(k) \quad \forall l. \end{cases} \quad (20)$$

It is clear that all these constraints are linear in  $\tilde{x}(k)$  and  $\tilde{u}(k)$ , and we end up with the linear inequality constraint:

$$A_c \begin{bmatrix} \tilde{x}(k) \\ \tilde{u}(k) \end{bmatrix} \leq b_c(k), \quad (21)$$

where  $b_c(k)$  contains all known elements of (20) at cycle  $k$ .

Let us now briefly show that any optimal solution of (18) subject to (20) satisfies (19). In order to show this we show that it is not possible for the optimal solution to have all strict inequalities in (20), i.e., at least one of the three inequalities holds with equality. Indeed, since the right-hand side of the first and third inequalities are known, since the matrices  $\tilde{A}_v$  can be turned into a strictly lower triangular matrix, and since the coefficient  $c_e$  of  $\tilde{x}(k)$  in  $J(k)$  is positive, then by contradiction it holds that if all inequalities in (20) hold in the strict sense then  $\tilde{x}(k)$  cannot be optimal.

So we have a linear objective function (18) that has to be minimised subject to the linear constraints (21) over real variables  $\tilde{x}(k)$  and binary variables  $\tilde{u}(k)$ . Hence, we finally end up with an MILP.

## 6. EXAMPLE

Now we compare the proposed control method for a simple railway traffic network with different delay scenarios. For all simulations the prediction horizon was set to  $N_p = 6$ . The weights in (18) were set equal to 1 for all departure components of  $\tilde{x}(k)$  and equal to  $10^{-6}$  for all arrival components. The penalty on the controls was set equal to  $10^{-3}$ . The network used to run the simulations is given in Figure 2.

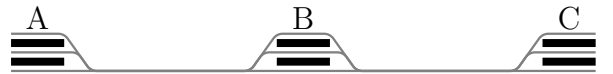


Fig. 2. A simple railway network.

The network has three stations denoted as A, B, and C. In this example we are only interested in trains in the direction A-C and so we consider only a single track. At all three stations there are multiple platforms which means there is sufficient capacity for take over operations. The period of the timetable is  $T = 60$  [min]. During every period there are four local trains and four intercity trains running from station A to C. Both types of train make a stop at station B, the local train also stops at some intermediate stations. We assume that these intermediate stations have no overtaking possibility, and are therefore omitted in the analysis. The corresponding timetable is given in Table 1.

The minimum dwell time of all trains is fixed at  $s_j(k) = 1$  [min]  $\forall j$ . The minimum headway time of all trains is fixed

Table 1. Timetable (d=departure, a=arrival).

		Line 1: Local train				Line 2: Intercity			
Train number		101	102	103	104	201	202	203	204
Station A	d	00	15	30	45	09	24	39	54
Station B	a	15	30	45	00	18	33	48	03
	d	23	38	53	08	20	35	50	05
Station C	a	35	50	05	20	27	42	57	12

at  $h_{ij}(k) = 3$  [min]  $\forall i, j$ . The minimum running time is always one minute less than the scheduled running time.

Three delay scenarios are simulated.

In the first scenario intercity train 201 with scheduled departure time 8:09 at station A has a delay less than 6 minutes. In this scenario changing the departure order with the proposed approach results in no delay reduction.

In the second scenario intercity train 201 at station A has a delay of 12 minutes. By not changing the order of the trains, the sum of delays becomes 51 [min]. Using the proposed approach intercity train 201 is rescheduled behind local train 102 on track A-B and behind local train 101' on track B-C. The total delay now reduces to 30 [min] resulting in a delay reduction of 21 [min]. The corresponding time-distance diagram is given in Figure 3.

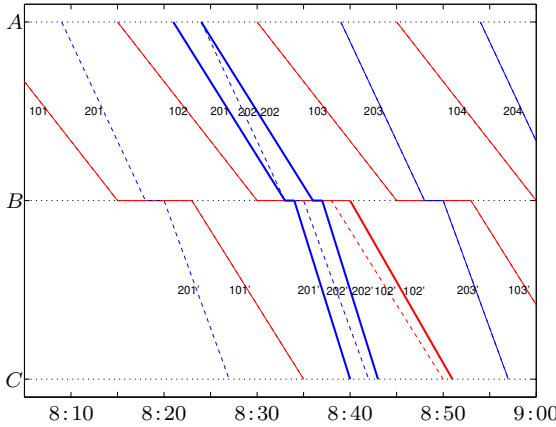


Fig. 3. Time-distance diagram for scenario 2. The bold lines represent rescheduled trains, whereas the dashed lines represent the original train paths.

In the third scenario intercity train 201 with scheduled departure time 8:09 at station A has a delay of 22 minutes. If the order of the trains is not changed, the sum of delays becomes 159 [min]. Using the proposed approach intercity train 201 is rescheduled behind train 102 and train 202 on track A-B and behind train 101', train 202', and train 102' on track B-C. The total delay reduces to 51 [min] and thus a delay reduction of 108 [min] is achieved. The corresponding time-distance diagram is given in Figure 4.

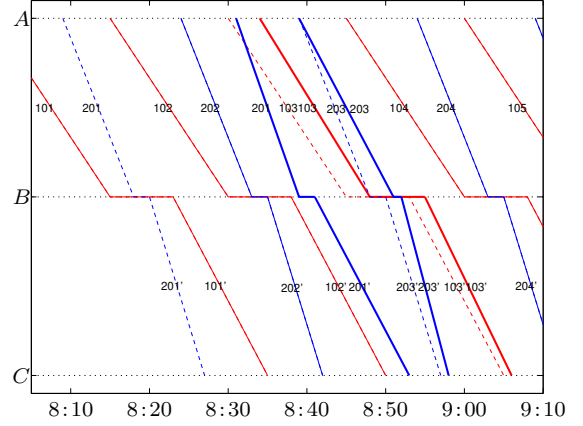


Fig. 4. Time-distance diagram for scenario 3. The bold lines represent rescheduled trains, whereas the dashed lines represent the original train paths.

## 7. CONCLUSIONS AND FUTURE WORK

We have presented an approach to optimally reschedule trains on a railway network based on a permutation method. We have modelled the system based on the switching max-plus framework and showed how the control problem can be recast as a mixed-integer linear programming problem. Compared to previous results we have extended the control actions to change the departure and arrival order of non-subsequent trains. For a simple railway network we have shown that by optimally rescheduling trains using the proposed approach delays can be substantially reduced.

Further research will concentrate on reducing the complexity of the mixed-integer linear programming problem by considering only a reduced number of relevant decision variables based for example on the delay propagation algorithm proposed in (Goverde, 2010).

## REFERENCES

- Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J. (1992). *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Wiley.
- Goverde, R.M.P. (2007). Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B*, 41(2), 179–201.
- Goverde, R. (2010). A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(3), 269–287.
- Heidergott, B. and de Vries, R. (2001). Towards a (max,+) control theory for public transportation networks. *Discrete Event Systems: Theory and Applications*, 11(4), 371–398.
- van den Boom, T. and 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, T. and De Schutter, B. (2007). On a model predictive control algorithm for dynamic railway network management. In I. Hansen, A. Radtke, J. Pachl, and E. Wendler (eds.), *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis (RailHannover2007)*. Hannover, Germany.