Technical report 07-009

On a model predictive control algorithm for dynamic railway network management[*]

T.J.J. van den Boom and B. De Schutter

On a model predictive control algorithm for dynamic railway network management

Ton J.J. van den Boom and Bart De Schutter

Delft University of Technology, Delft Center for Systems and Control, Mekelweg 2, 2628
CD Delft, The Netherlands, e-mail: a.j.j.vandenboom@tudelft.nl,
b.deschutter@dcsc.tudelft.nl

Abstract

In this paper we discuss a model predictive control method for dynamic traffic management of railway networks. The main aim of the predictive controller is to recover from delays in an optimal way by breaking connections and changing the departure of trains (at a cost). To model the railway system we use a switching max-plus-linear system description. We define the model predictive control design problem for the railway network, and we show that the problem can be recast into a mixed integer linear programming problem. This problem can be solved using existing solvers for MILP problems, or with a genetic algorithm or tabu search algorithm. We also apply the control strategy to a model of the Dutch railway network.

Keywords: Railway network management, switching max-plus-linear models, model predictive control.

1 Introduction

In recent years a lot of research effort has been oriented towards the design of timetables that are robust against propagation of delays in the network, caused by technical failures, fluctuation of passenger volumes, measures of railway personnel and weather influence [19, 20, 23, 24]. In this paper we concentrate on the operational-level management, and design a feedback controller that takes the most effective actions, based on measurements of the actual train positions. The measures we can take are changing the train speed, breaking train connections, or changing the order of trains.

From [4, 6, 14, 15, 22] we know that a railway network with rigid connection constraints and a fixed routing schedule can be modeled using max-plus-linear models. A max-plus-linear model is 'linear' in the max-plus algebra [2], which has maximization and addition as its basic operations. Max-plus-linear systems can be characterized as discrete event systems in which only synchronization and no concurrency or choice occurs.

Note that in the railway context, synchronization means that some trains should give pre-defined connections to other trains, and a fixed routing means that the order of departure is fixed. However, in the case of large delays, it is sometimes better — from a global performance viewpoint — to break a connection or to reschedule the order of trains, and to let a train depart anyway. In this way we prevent an accumulation of delays in the network. Of course, missed connections should lead to a penalty due to dissatisfied passengers. In [10, 11] we have considered the control of railway networks using breaking connections

only as control measure. In [25] we have extended the control handles and rescheduled the trains by breaking connections as well as changing train order. In this paper we will model a controlled railway system using the *switching* max-plus-linear system description of [27]. In this description we use a number of MPL models, each model corresponds 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 break train connections and to change the order of trains. In this paper we define a control algorithm to optimize the performance of the network, and we show that the resulting optimization problem can be solved as a mixed integer problem or a mixed integer linear programming problem. Although these problems are in general NP-hard, recently several efficient solvers have become available. The management algorithm will be applied to a simulation model of the Dutch railway network. Computational experiments show that the proposed genetic algorithm approach yields good results.

2 Model

Consider a railway operations system, which follows a schedule with period T. In nominal operation mode, we assume that all the trains follow a pre-scheduled route, with fixed train order and pre-defined connections. If for some reason we have to break connections or change the train order, we will operate in a perturbed mode. With every new schedule we can associate a perturbed mode. First we will discuss the nominal operation.

2.1 Nominal operation

Consider a railway operations system which is operating in nominal operation mode.

Each track of the railway network has a number and a train allocated to it. For the sake of simplicity we will say '(virtual) train j' to denote the (physical) train on track j, and 'station j' to denote the station at the beginning of track j (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 may actually be the same physical track (corresponding to different trains using the same track). This means that the number of tracks is usually smaller than n. Let $d_j(k)$ be the time instant at which train j departs from its departure station for the kth time, and let $a_j(k)$ be the time instant at which train j arrives from its arrival station for the kth time Let $r_j(k)$ be the departure time for this train according to the time schedule. Let $p_i(k)$ be the predecessor track of train i, and let $\mathcal{C}_j(k)$ be the set of trains to which the kth train j gives a connection. Let $\mathcal{F}_j(k)$ be the set of trains that move over the same track as train j, in the same direction as train j, and are scheduled behind train j. Let $\mathcal{W}_j(k)$ be the set of trains that move over the same track as train j, in the opposite direction of train j, and are scheduled behind train j. Furthermore, let $t_j(k)$ be the traveling time on track j, define a minimum connection time $c_{ij}^{\min}(k)$ for passengers to get from train j to train i for each train $j \in \mathcal{C}_i(k)$ and define a minimum stopping time $s_j^{\min}(k)$ of train j at station j to allow passengers to get off or on the train. Finally, define a minimum separation time $f_j^{\min}(k)$ between two different trains moving over the same track and in the same direction as train j, and a minimum separation time $w_j^{\min}(k)$ between two different trains moving over the same track and in the opposite direction. same track and in the opposite direction.

Now we have the following constraints for the kth departure time $x_i(k)$ of train i:
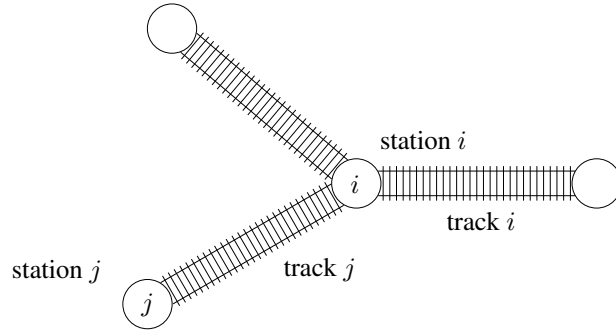
Figure 1: A part of a railway network.

- *Time schedule constraint:*

$$d_i(k) \geqslant r_i(k) \ .$$

- *Running constraint:* This gives the relation between the departure and arrival time of a train. For train i we have

$$a_i(k) = d_i(k - \delta_i^*(k)) + t_i(k)$$

where $\delta_i^*(k)$ is equal to 0 if train arrives at its destination in the same cycle as its departure, and $\delta_i^*(k)$ is equal to n is it arrives n cycles later than the departure.

- *Continuity constraints:* This constraint synchronizes two trains that are 'physically' the same train. For train $j = p_i(k)$ we have

$$d_i(k) \geqslant a_j(k - \delta_{ij}^*(k)) + s_j^{\min}(k)$$

where $\delta_{ij}^*(k)$ is equal to n if the $(k-n)$th train j continues as the kth train i, and 0 if the kth train j continues as the kth train i.

- *Connection constraints:* This constraint synchronizes two trains that have to make a connection. For each train $i \in \mathcal{C}_j(k)$ we have

$$d_i(k) \geqslant a_j(k - \delta_{ij}^*) + c_{ij}^{\min}(k)$$

where the role of δ_{ij}^* is similar as for the continuity constraint, so $\delta_{ij}^* = 1$ if the $(k-1)$th train j gives a connection to the kth train i, and $\delta_{ij}^* = 0$ if the kth train j gives a connection to the kth train i.

- *Follow constraints:* This constraint synchronizes two subsequent trains on the same track moving in the same direction . For each train $i \in \mathcal{F}_j(k)$ we have constraints for both departure and arrival

$$d_i(k) \geqslant d_j(k - \delta_{ij}^*) + f_j^{\min}(k) \qquad a_i(k) \geqslant a_j(k - \delta_{ij}^*) + f_j^{\min}(k)$$

(δ_{ij}^* is defined similarly as above).

- *Wait constraints:* This constraint synchronizes two trains on the same track moving in opposite direction. For each train $i \in \mathcal{W}_j(k)$ we have

$$d_i(k) \geqslant a_j(k - \delta_{ij}^*) + w_j^{\min}(k)$$

  (δ_{ij}^* is defined similarly as above).

- *Crossing constraint:* This constraint synchronizes two trains at a crossing point. A crossing point can be seen as a short track that has to be used by two consecutive trains. The first train must have left the crossing before the second train is allowed to enter the crossing. The constraint will therefore have the same form as the wait constraint.

In general, all δ_{ij}^*'s may depend on k. However, for the sake of simplicity, we only consider constant δ_{ij}^*'s with a value that is either 0 or 1 in this paper. Since we let a train depart as soon as all connection conditions are satisfied, we have

$$d_i(k) = \max\big(r_i(k), (a_{p_i(k)}(k - \delta_{ip_i(k)}^*) + s_{p_i(k)}^{\min}(k)), \max_{j \in \mathcal{C}_i(k)} (a_j(k - \delta_{ij}^*) + c_{ij}^{\min}(k)),$$
$$\max_{l \in \mathcal{F}_i(k)} (a_l(k - \delta_{il}^*) + f_l^{\min}(k)), \max_{m \in \mathcal{W}_i(k)} (a_m(k - \delta_{im}^*) + w_m^{\min}(k)))) \tag{1}$$

Note that in a undisturbed, well-defined time schedule the term $r_i(k)$ in (1) will be the largest. However, if due to unforeseen circumstances (an incident, a late departure, etc.) one of the trains ($p_i(k)$,l or m) has a delay the corresponding term can become larger than the others, then train i will depart later than the scheduled departure time $r_i(k)$ and will therefore also be delayed. Now let us consider a network with n trains and define the vectors

$$x(k) = \begin{bmatrix} d_1(k) \\ \vdots \\ d_n(k) \\ a_1(k) \\ \vdots \\ a_n(k) \end{bmatrix} \in \mathbb{R}^{2n} \quad , \quad r(k) = \begin{bmatrix} r_1(k) \\ \vdots \\ r_n(k) \\ -\infty \\ \vdots \\ -\infty \end{bmatrix}$$

(so $r_i(k) = -\infty$ for $n + 1 \leq i \leq 2n$). By defining the appropriate matrix $A_m \in \mathbb{R}^{2n \times 2n}$, $m = 0, \ldots, m_{\max}$, (where $m_{\max} = \max(\delta_{ij}^*)$ we can rewrite equation (1) as:

$$x_i(k) = \max\left(r_i(k), \max_{j,m} \left(x_j(k-m) + [A_m]_{i,j}\right)\right) \tag{2}$$

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

Now we introduce some notation from max-plus algebra. Define $\varepsilon = -\infty$ and $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$. The max-plus-algebraic addition ($\oplus$) and multiplication ($\otimes$) are defined as follows [2]:

$$x \oplus y = \max(x, y) \qquad x \otimes y = x + y$$

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

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

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

for $A, B \in \mathbb{R}_{\varepsilon}^{m \times n}$, $C \in \mathbb{R}_{\varepsilon}^{n \times p}$. The matrix $\mathcal{E}$ is the max-plus-algebraic zero matrix: $[\mathcal{E}]_{ij} = \varepsilon$ for all i, j.

In max-plus notation, equation (2) becomes

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

and in matrix-notation we obtain

$$x(k) = A_0 \otimes x(k) \oplus A_1 \otimes x(k-1) \oplus A_{m_{\max}} \otimes x(k - m_{\max}) \oplus r(k)$$
$$= \Big(\bigoplus_{m=0}^{m_{\max}} A_m \otimes x(k - m) \Big) \oplus r(k) \tag{3}$$

2.2 Perturbed operation

In the nominal operation we have assumed that some trains should give pre-defined connections to other trains, and 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 will consider the switching between different operation modes, where each mode corresponds to a different set of pre-defined or broken connection and a specific order of train departures. We allow the system to switch between different modes, allowing us to break train connections and to change the order of trains. Note that any broken connection or change of train order leads to a new model, similar to the nominal equation (3), but now with adapted system matrices $(A(\ell))$ for the ℓ-th model. We have the following system equation for the perturbed operation:

$$x(k) = \Big(\bigoplus_{m=m_{\min}}^{m_{\max}} A_m(\ell(k)) \otimes x(k - m) \Big) \oplus r(k) \tag{4}$$

Usually $m_{\min} = 0$. However, in perturbed operation it may occasionally happen that a delayed train of the kth cycle is rescheduled behind a train in the $(k + 1)$th cycle. In that case we will have $m_{\min} = -1$. The nominal behavior of the system, as defined by the nominal timetable is related to the nominal model $\ell = 0$, with corresponding system matrices $A_m(0)$.

3 The railway control problem

3.1 Timing aspects

Switching max-plus-linear systems are different from conventional time-driven systems in the sense that the event counter k is not directly related to a specific time. A time instant t

in cycle k (so $(k-1)T \le t < kT$), some of the components of $x(k)$ may already be known while other components of $x(k)$ may still lie in the future (Recall that $x(k)$ contains the time instants at which the depart or arrive in the kth cycle). Therefore, we will now present a method to address the timing issues in control of switching MPL systems.

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

Consider time instant t and let cycle $(k-1)$ be the last completed cycle, so $x(k-1) \le t$. We may have measurements of trains in cycle k that have already departed or arrived at time t. We will denote them as $a_{\text{past}}(k)$ and $a_{\text{past}}(k)$ respectively. Sometimes there is information available about the estimated traveling time for trains that have not yet arrived at their destination at time t. With this information we can make an estimation $\hat{t}_{\text{est}}(k|t)$ (with the same dimension as $t_(k)$) of the future traveling times. If no further information is available on a specific traveling time we take the nominal traveling time $[\hat{t}_{\text{est}}(k|t)]_i = t_{i,\text{nom}}$. These values $\hat{t}_{\text{est}}(k|t)$ can be substituted in the matrices $A_m(\ell(k))$ to make the future system description as accurate as possible.

3.2   Control problem

Next we have to define the set $\mathcal{U}(k|t)$ of possible future control actions (i.e. breaking connections or changing train order). Certain control actions are not feasible any more (e.g. if a connection has been broken in the past and the connecting train has already departed, it is impossible to 'repair' this connection.). We define the vector $u(k|t) \in \mathcal{U}(k|t)$, where each element corresponds to a specific control action, so a specific (scheduled) connection or specific (scheduled) train order. We assume $u(k|t)$ to be binary, where $u_i(k|t) = 0$ corresponds to the nominal case, and $u_i(k|t) = 1$ to the perturbed case (the connection is broken or the order of two trains is switched, see also Section 5).

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

$$\min_{\{u(k|t),u(k+1|t),u(k+2|t),\ldots\}} J(k|t) \tag{5}$$

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

$$J(k,t) = \sum_{j=0}^{N_{\text{p}}} \sum_{i=1}^{n} Q_i\, \hat{e}_i(k+j|t) + \sum_{\ell=1}^{n_u} R_\ell\, u_\ell(k+j|t) \tag{6}$$

where N_{p} is the prediction horizon, $\hat{e}(k+j|t)$ is the vector with the expected delays ($\hat{e}_i(k+j|t) = \hat{x}_i(k+j|t) - r_i(k+j) \ge 0$), and Q, R are weighting matrices. The first term of (6) is related to the sum of all predicted delays, and the second term denotes the penalty for all broken connections and switched train orders during cycle $k+j$.

To compute the predictions of $\hat{x}(k+j|t)$ we make use of the fact that at time t we have $a_{\text{past}}(k|t)$, $a_{\text{past}}(k|t)$, and $\hat{t}_{\text{est}}(k+j|t)$ available and using that we can determine the estimates $\hat{A}_m(\ell(k+j|t))$ of all future $A_m(\ell(k+j))$. Now $\hat{x}(k+j|t)$ for $j \ge 1$ can be found by successive substitution

$$\hat{x}(k+j|t) = \Big(\bigoplus_{m=m_{\min}}^{m_{\max}} \hat{A}_m(\ell(k+j-1|t)) \otimes \hat{x}(k+j-m|t) \Big) \oplus r(k+j) \tag{7}$$

6

In principle we have all elements to solve the optimal control problem (5). Note that if the railway network is well-defined and there is some margin in the schedule, there will always be an integer N such that in the nominal case ($u(k+j|t) = 0$ for all $j \geq 0$) the delays will have vanished ($\hat{e}(k+j|t) = 0$ for all $j \geq 0$). By choosing $N_{\mathrm{p}} = N$ in the performance index (6) 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 optimization problem is drastically reduced.

4 Affine models

If the model is affine in the control parameters, the model predictive control problem can be recast into a mixed integer linear programming (MILP) problem using techniques that are similar to the ones used in [3, 9]. We start by showing that in the case of breaking connections and changing departure order the matrices $\hat{A}_m(\ell(k))$ can be written in an affine form:

$$\hat{A}_m(\ell(k)) = \hat{A}_m(0) + \sum_{i=1}^{n_u} \hat{A}_m^i \, u_i(k) \; , \tag{8}$$

In subsection 4.3 we proof that the problem becomes a MILP problem.

4.1 Breaking connections

Consider the case where variable $u_l(k)$ is related to the connection of train j to train i, with nominal connection constraint

$$d_i(k) \geqslant a_j(k - \delta_{ij}^*) + c_{ij}^{\min}(k)$$

This means that in the nominal case we have

$$[A_{\delta_{ij}^*}(0)]_{p,q} = \begin{cases} c_{ij}^{\min}(k) & \text{for } \{p,q\} = \{i, j+n\} \\ \epsilon & \text{elsewhere} \end{cases}$$

To break the constraint we replace the above connection constraint by

$$d_i(k) \geqslant a_j(k - \delta_{ij}^*) + c_{ij}^{\min}(k) - \beta \, u_i(k)$$

where β is a large number such that for $u_l(k) = 1$ the constraint is satisfied over the whole time schedule. (so if $T_{\max}$ and T_{min} are the maximum and minimum time instants in the time schedule we make sure that $\beta \ll T_{\max} - T_{min}$). Now the matrix $A_{\delta_{ij}^*}(l)$ is defined as:

$$[A_{\delta_{ij}^*}(l)]_{p,q} = \begin{cases} -\beta & \text{for } \{p,q\} = \{i, j+n\} \\ \epsilon & \text{elsewhere} \end{cases}$$

4.2 Changing departure order

Consider the case where variable $u_l(k)$ is related to the order of two trains j and i moving over the same track in the same direction, with nominal following constraints

$$d_i(k) \geqslant d_j(k - \delta_{ij}^*) + f_j^{\min}(k) \tag{9}$$

$$a_i(k) \geqslant a_j(k - \delta_{ij}^*) + f_j^{\min}(k) \tag{10}$$

To be able to change the order of the trains i and j we replace these constraints by

$$d_i(k) \geqslant d_j(k - \delta_{ij}^*) + f_j^{\min}(k) - \beta\, u_l(k) \tag{11}$$

$$a_i(k) \geqslant a_j(k - \delta_{ij}^*) + f_j^{\min}(k) - \beta\, u_l(k) \tag{12}$$

where β is a large number (see previous subsection), and introduce new constraints

$$d_j(k) \geqslant d_i(k + \delta_{ij}^*) - \beta + (f_i^{\min}(k) + \beta)\, u_l(k) \tag{13}$$

$$a_j(k) \geqslant a_i(k + \delta_{ij}^*) - \beta + (f_i^{\min}(k) + \beta)\, u_l(k) \tag{14}$$

so that for $u_l(k) = 0$ the constraints (11–12) are relevant and the constraints (13–14) are always satisfied, where for $u_l(k) = 1$ the constraints (13–14) are relevant and the constraints (11–12) are always satisfied.

This means that for $\delta_{ij}^* \neq 0$ the nominal matrices $A_{\delta_{ij}^*}(0)$ and $A_{\delta_{ij}^*}(0)$ are given by

$$[A_{\delta_{ij}^*}(0)]_{p,q} = \begin{cases} f_j^{\min}(k) & \text{for } \{p,q\} = \{i,j\} \\ f_j^{\min}(k) & \text{for } \{p,q\} = \{i+n, j+n\} \\ \epsilon & \text{elsewhere} \end{cases}$$

$$[A_{-\delta_{ij}^*}(0)]_{p,q} = \begin{cases} -\beta & \text{for } \{p,q\} = \{j,i\} \\ -\beta & \text{for } \{p,q\} = \{j+n, i+n\} \\ \epsilon & \text{elsewhere} \end{cases}$$

and for $\delta_{ij}^* = 0$ the nominal matrix $A_0(0)$ is given by:

$$[A_0(0)]_{p,q} = \begin{cases} f_j^{\min}(k) & \text{for } \{p,q\} = \{i,j\} \\ f_j^{\min}(k) & \text{for } \{p,q\} = \{i+n, j+n\} \\ -\beta & \text{for } \{p,q\} = \{j,i\} \\ -\beta & \text{for } \{p,q\} = \{j+n, i+n\} \\ \epsilon & \text{elsewhere} \end{cases}$$

Now for $\delta_{ij}^* \neq 0$ the matrices $A_{\delta_{ij}^*}(l)$ and $A_{\delta_{ij}^*}(l)$ are given by

$$[A_{\delta_{ij}^*}(l)]_{p,q} = \begin{cases} -\beta & \text{for } \{p,q\} = \{i,j\} \\ -\beta & \text{for } \{p,q\} = \{i+n, j+n\} \\ \epsilon & \text{elsewhere} \end{cases}$$

$$[A_{-\delta_{ij}^*}(l)]_{p,q} = \begin{cases} f_i^{\min}(k) + \beta & \text{for } \{p,q\} = \{j,i\} \\ f_i^{\min}(k) + \beta & \text{for } \{p,q\} = \{j+n, i+n\} \\ \epsilon & \text{elsewhere} \end{cases}$$

and for $\delta_{ij}^* = 0$ the matrix $A_0(l)$ is given by

$$[A_0(l)]_{p,q} = \begin{cases} -\beta & \text{for } \{p,q\} = \{i,j\} \\ -\beta & \text{for } \{p,q\} = \{i+n, j+n\} \\ f_i^{\min}(k) + \beta & \text{for } \{p,q\} = \{j,i\} \\ f_i^{\min}(k) + \beta & \text{for } \{p,q\} = \{j+n, i+n\} \\ \epsilon & \text{elsewhere} \end{cases}$$

4.3 The mixed integer linear programming problem (MILP)

Now the model predictive problem above can be recast into a mixed integer linear programming problem (MILP). We will now outline the main ideas behind this transformation. Define the vectors

$$
\tilde{x}(k) = \begin{bmatrix} x(k) \\ \vdots \\ x(k+N_\mathrm{p}-2) \\ x(k+N_\mathrm{p}-1) \end{bmatrix}, \quad \tilde{u}(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N_\mathrm{p}-2) \\ u(k+N_\mathrm{p}-1) \end{bmatrix}, \quad \tilde{z}(k) = \begin{bmatrix} x(k-1) \\ \vdots \\ x(k-m_\mathrm{max}+1) \\ x(k-m_\mathrm{max}) \end{bmatrix},
$$

$$
\tilde{\ell}(k) = \begin{bmatrix} \ell(k) \\ \vdots \\ \ell(k+N_\mathrm{p}-2) \\ \ell(k+N_\mathrm{p}-1) \end{bmatrix}, \quad \tilde{r}(k) = \begin{bmatrix} r(k) \\ \vdots \\ r(k+N_\mathrm{p}-2) \\ r(k+N_\mathrm{p}-1) \end{bmatrix},
$$

Note that by defining

$$
\tilde{A}(\tilde{\ell}(k)) = \begin{bmatrix} A_0(\ell(k)) & A_{-1}(\ell(k)) & \cdots & A_{-N_\mathrm{p}}(\ell(k)) \\ A_1(\ell(k+1)) & A_0(\ell(k+1)) & \ddots & A_{1-N_\mathrm{p}}(\ell(k)) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{E} & \mathcal{E} & \cdots & A_{m_\mathrm{max}}(\ell(k+N_\mathrm{p})) \end{bmatrix}
$$

and

$$
\tilde{B}(\tilde{\ell}(k)) = \begin{bmatrix} A_1(\ell(k)) & \cdots & A_{m_\mathrm{max}-1}(\ell(k)) & A_{m_\mathrm{max}}(\ell(k)) \\ A_2(\ell(k+1)) & \cdots & A_{m_\mathrm{max}}(\ell(k+1)) & \mathcal{E} \\ \vdots & & \vdots & \vdots \\ A_{m_\mathrm{max}}(\ell(k+m_\mathrm{max}-1)) & \cdots & \mathcal{E} & \mathcal{E} \\ \vdots & & \vdots & \vdots \\ \mathcal{E} & \cdots & \mathcal{E} & \mathcal{E} \end{bmatrix}
$$

where $A_m(\ell(k+j)) = \mathcal{E}$ for $m < m_{min}$ and for $m > m_\mathrm{max}$, we can write

$$
\tilde{x}(k) = \tilde{A}(\tilde{\ell}(k)) \otimes \tilde{x}(k) \oplus \tilde{B}(\tilde{\ell}(k)) \otimes \tilde{z}(k) \otimes \tilde{r}(k) \tag{15}
$$

Note that due to (8) the matrices $\tilde{A}(\tilde{\ell}(k))$ and $\tilde{B}(\tilde{\ell}(k))$ will be affine in $\tilde{u}$, and there exist matrices $\tilde{A}^i$ and $\tilde{B}^i$ such that:

$$
\tilde{A}(\tilde{u}(k)) = \tilde{A}^0 + \sum_{i=1}^{n_u} \tilde{A}^i \tilde{u}_i(k+j|k) \ ,
$$

$$
\tilde{B}(\tilde{u}(k)) = \tilde{B}^0 + \sum_{i=1}^{n_u} \tilde{B}^i \tilde{u}_i(k+j|k) \ ,
$$

The objective function J is linear in $\tilde{u}$ and $\tilde{x}$, and can be written as:

$$
J(k) = c_e^T \, \tilde{x}(k) + c_u^T \, \tilde{u}(k) \tag{16}
$$

The max-plus equation (7) can be transformed into a system of mixed-integer linear inequalities. Equation (15) can be written as

$$\tilde{x}_i(k) = \max\left(\tilde{r}_i(k), \max_j \tilde{x}_j(k) + [\tilde{A}(\tilde{u})]_{i,j}, \max_j \tilde{z}_j(k) + [\tilde{B}(\tilde{u})]_{i,j}\right),$$

which can be transformed into

$$\tilde{x}_i(k) \geq \tilde{r}_i(k)$$

$$\tilde{x}_i(k) \geq \tilde{x}_j(k) + [\tilde{A}^0]_{i,j} + \sum_{p=1}^{n_u} [\tilde{A}^p]_{i,j}\tilde{u}_p(k) \quad , \quad \forall j$$

$$\tilde{x}_i(k) \geq \tilde{z}_l(k) + [\tilde{B}^0]_{i,l} + \sum_{p=1}^{n_u} [\tilde{B}^p]_{i,l}\tilde{u}_p(k) \quad , \quad \forall l$$

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

$$A_c \left[\begin{array}{c} \tilde{x}(k) \\ \tilde{u}(k) \end{array} \right] \leq b_c(k) \tag{17}$$

So we have a linear objective function (16) that has to be minimized subject to the linear constraints (17) over real variables $\tilde{x}(k)$ and binary variables $\tilde{u}(k)$. Hence, we finally end up with an MILP.

5 Solving the MILP problem

The MILP problem as derived in the previous section can be solved using one of the several existing commercial and free solvers for MILP problems (such as, e.g., CPLEX, Xpress-MP, GLPK, lp_solve, etc. — see [1, 21] for an overview). However,if the network is too large, the MILP solvers will not converge fast enough and we need other solvers, e.g. genetic algorithms [8] or with tabu search [17], to efficiently solve our integer optimal control problem. An important issue then is to generate a good initial solution for the optimization problem.

To find a good initial guess for the integer optimization we first solve an easier problem, in which we structure the input signal. This is done by defining a decision mechanism, where we use thresholds on (expected) delays to decide whether a connection should be broken or train orders should be switched. First consider the case where variable $u_l(k)$ is related to the connection of train j to train i, with nominal connection constraint

$$d_i(k) \geqslant a_j(k - \delta_{ij}^*) + c_{ij}^{\min}(k)$$

and let $r_i(k) > t$. Define $\hat{a}_j(k - \delta_{ij}^*|t)$ as the expected arrival-time of train j. Now we choose

$$\begin{cases} u_l(k) = 0 & \text{if } \hat{a}_j(k-\delta_{ij}^*|t)+c_{ij}^{\min}(k)-r_i(k) \leq \tau \\ u_l(k) = 1 & \text{otherwise,} \end{cases}$$

where τ is a non-negative threshold. Next consider the case where variable $u_l(k)$ is related to the order of two trains j and i moving over the same track in the same direction, with nominal following constraint

$$d_i(k) \geqslant d_j(k - \delta_{ij}^*) + f_j^{\min}(k)$$

10

and let $d_i(k) \geq t$ (that means that at time t train i has not departed yet). Now we choose

$$\begin{cases} u_l(k) = 0 & \text{if } \hat{d}_j(k-\delta_{ij}^*|t)+f_j^{\min}(k)-r_i(k) \leq \phi \\ u_l(k) = 1 & \text{otherwise,} \end{cases}$$

where ϕ is a non-negative threshold. Finally consider the case where variable $u_l(k)$ is related to the order of two trains j and i moving over the same track in the same direction, with nominal waiting constraint

$$d_i(k) \geqslant a_j(k - \delta_{ij}^*)$$

and let $r_i(k) > t$. Now we choose

$$\begin{cases} u_l(k) = 0 & \text{if } \hat{a}_j(k - \delta_{ij}^*|t) - r_i(k) \leq \omega \\ u_l(k) = 1 & \text{otherwise,} \end{cases}$$

where $\hat{a}_j(k - \delta_{ij}^*|t)$ is the expected arrival-time and ω is a non-negative threshold. In this structured-input case we end up with the minimization of (6) using the three parameters, giving us a non-linear optimization problem over the variables (τ, ϕ, ω). In the worked example in the next Section we first optimize over the structured inputs, and use the resulting sequence $u(k + j|t)$ as an initial value for the general case, solved with a genetic algorithm.

6 Example: The Dutch railway system

We consider a simulation example of a simplified version of the Dutch railway system (see Figure 2). The Dutch railway system operates a periodic timetable with a cycle time of 1 hour. There are 3 types of trains running in the network: intercity trains, interregional trains and local trains. In this example we only consider the system of intercity and interregional trains, which means that the network consists of 40 stations, 110 tracks, 164 trains, 381 train movements per hour, 271 follow constraints, and 67 connection constraints.

Each train departs as soon as all the relevant connections are guaranteed (except for connections that are broken), the passengers have gotten the opportunity to change over, and the earliest departure time indicated in the timetable has passed.

We obtain a state space description of the railway system in the form of (2) for nominal operation, and (4) for perturbed operation where $x(k)$ has 381 states. The criterion function is given by (6) where $Q = I$, and R is a diagonal matrix with $R_{\ell,\ell} = 5$ when u_ℓ is related to a connection constraint and $R_{\ell,\ell} = 10$ when u_ℓ is related to a follow constraint. We solve the optimal control problem (5) for the structured input case and the general case (without structuring). In the last optimization we use the result of the structured input as an initial value to start the optimization. We assume the system is at nominal schedule for $k < 0$ and we introduce random delays in the running times of 18 trains in the cycles $k = 1, 2, 3$. The maximum delay is 19.9 min, the minimum delay is 0.12 min, and the average delay is 11.9 min. For every cycle k we first optimize the threshold values (τ, ϕ) (there are no single tracks taken into consideration), and compute the corresponding optimal structured input signal $u_{\text{structured}}(k + j|t)$, $j \geq 0$. Subsequently we optimize the (unstructured) input signal $u(k + j|t)$ with a genetic algorithm, using the earlier computed sequence $u_{\text{structured}}(k + j|t)$ as an initial value.

In Figure 3 the maximum delay $e_{\max}(k) = \max(e(k))$ in each cycle k is given for the optimal controlled case, and for the uncontrolled case (so $u(k + j|t) = 0$ for all $j > 0$). We see that the delay in the controlled case decays much faster than the uncontrolled case.

11

Figure 2: Dutch railway network

7 Discussion

We have presented a control design method for a railway network. The control action consists in breaking certain connections or changing the order of departure to prevent delays from accumulating. These control moves can only be done at a certain cost. We have shown that the resulting optimization problem is a mixed integer linear programming (MILP) problem, that can be solved using existing commercial and free solvers for MILP problems, or by other integer optimization methods, for example genetic algorithms or tabu search.

Good initial values for the integer optimization are obtained by first solving a low-dimensional real-valued optimization problem using a structured input sequence. This structured input sequence is based on a decision mechanism, where we use thresholds on (expected) delays to decide whether a connection should be broken or the order of the trains should be switched.

Due to the use of a receding horizon this method can be used in on-line applications and it can deal with (predicted) changes in the system parameters. So if we can predict the delays that will occur due to an incident or to works, then we can include this information when determining the optimal control input for the next cycles of the operation of the network.
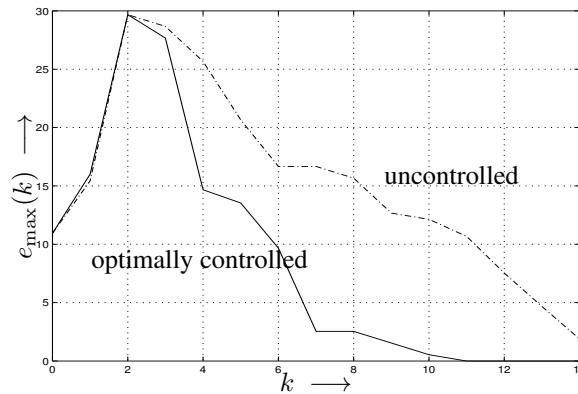
Figure 3: Maximum delay for uncontrolled and optimal controlled railway system

References

[1] A. Atamtürk and M. W. P. Savelsbergh. Integer-programming software systems. *Annals of Operations Research*, 2005. To appear.

[2] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity*. John Wiley & Sons, New York, 1992.

[3] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

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

[5] J.G. Braker. *Algorithms and Applications in Timed Discrete Event Systems*. PhD thesis, Department of Technical Mathematics and Informatics, Delft University of Technology, Delft, The Netherlands, December 1993.

[6] J.G. Braker. An extended algorithm for performance evaluation of timed event graphs. In *Proceedings of the 2nd European Control Conference*, pages 524–529, Groningen, The Netherlands, June 1993.

[7] R.A. Cuninghame-Green. *Minimax Algebra*, volume 166 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, Germany, 1979.

[8] L. Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[9] B. De Schutter, W.P.M.H. Heemels, and A. Bemporad. On the equivalence of linear complementarity problems. *Operations Research Letters*, 30(4):211–222, August 2002.

[10] B. De Schutter and T. van den Boom. Model predictive control for railway networks. In *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM'01)*, pages 105–110, Como, Italy, July 2001.

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

[12] B. De Schutter and T.J.J. van den Boom. Connection and speed control in railway systems — A model predictive control approach. In M. Silva, A. Giua, and J.M. Colom, editors, *Proceedings of the 6th International Workshop on Discrete Event Systems (WODES'02)*, pages 49–54, Zaragoza, Spain, October 2002.

[13] B. De Schutter and T.J.J. van den Boom. MPC for discrete-event systems with soft and hard synchronisation constraints. *International Journal of Control*, 76(1):82–94, 2003.

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

[15] P.R. de Waal, A. Overkamp, and J.H. van Schuppen. Control of railway traffic on a single line. In *Proceedings of the European Control Conference (ECC'97)*, Brussels, Belgium, paper 230, July 1997.

[16] F. Glover. Tabu search: A tutorial. *Interfaces*, 20(4):74–94, 1990.

[17] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Boston, 1997.

[18] F. Glover, E. Taillard, and D. de Werra. A user's guide to tabu search. *Annals of Operations Research*, 41:3–28, 1993.

[19] R.M.P. Goverde. *Punctuality of Railway Operations and Timetable Stability Analysis*. PhD thesis, Delft University of Technology, TRAIL Thesis Series T2005/10, Delft, The Netherlands, 2005.

[20] I. Hansen. Improving railway punctuality by automatic piloting. In *Proceedings IEEE Intelligent Transportation Systems*, pages 792–797, Oakland, CA, August 25-29 2001.

[21] J. Linderoth and T. Ralphs. Noncommercial software for mixed-integer linear programming. *Optimization Online*, December 2004. See http://www.optimization-online.org/DB_HTML/2004/12/1028.html.

[22] R. Minciardi, M. Paolucci, and R. Pesenti. Generating optimal schedules for an underground railway line. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 4082–4085, New Orleans, Louisiana, December 1995.

[23] L. Peeters. *Cyclic Railway Timetable Optimization*. PhD thesis, Erasmus University Rotterdam, ERIM PhD series in Management 22, Rotterdam, The Netherlands, 2003.

[24] Subiono. *On classes of min-max-plus systems and their applications*. PhD thesis, Delft University of Technology, TRAIL Thesis Series, T2000/2, Delft, The Netherlands, 2000.

[25] T.J.J. van den Boom and B. De Schutter. Modelling and control of railway networks. In *Proceedings of the American Control Conference 2004*, pages 5728–5733, Boston, MA, June/July 2004.

[26] T.J.J. van den Boom and B. De Schutter. Dynamic railway network management using switching max-plus-linear models. In *Proceedings of the 11th IFAC sysmposium on Control in Transportation Systems 2006*, Delft, the Netherlands, August 2006.

[27] T.J.J. van den Boom and B. De Schutter. Modelling and control of discrete event systems using switching max-plus-linear systems. *Accepted for publication in Control Engineering Practice*, Accepted November 2005, Final version January 2006.