Model predictive scheduling of semi-cyclic discrete-event systems using switching max-plus linear models

with an application in railway traffic management

Ton J.J. van den Boom

Delft Center for Systems and Control
Delft University of Technology

Email: A.J.J.vandenBoom@tudelft.nl

Thanks to Bart De Schutter, Bart Kersbergen, Gabriel Lopes, Lex Blenkers
Marenne van den Muijsenberg, Graziana Cavone

Contents

1st part: Scheduling of semi-cyclic discrete-event systems.

- Switching max-plus linear systems + Dynamic graphs.
- Examples of SMPL systems.
- Scheduling with SMPL systems.
- Model Predictive Scheduling.

2nd part: Application on Railway traffic management.

- Basic Railway traffic model.
- Distributed Model Predictive Scheduling.
- Disruption management.

Part 1: Scheduling of semi-cyclic discrete-event systems.

Scheduling is the process of deciding how to allocate a set of jobs to limited resources over time in such a way that one or more objectives are optimized.

Operational scheduling or *rescheduling* deals with adaptive on-line scheduling in response to the unexpected events.

Cyclic discrete-event system: Jobs appear in a repatative way.

Semi-cyclic discrete-event system: Changes in jobs and resources per cycle may occur.

Max-Plus Algebra

Define $\varepsilon = -\infty$ and $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$.

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

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

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

$$[A \odot B]_{ij} = [A]_{ij} + [B]_{ij}$$

Let $v \in \mathbb{B}_\varepsilon = \{0, \varepsilon\}$ be a max-plus binary variable.
The adjoint variable $\bar{v} \in \mathbb{B}_\varepsilon$ is defined as follows:

$$\bar{v} = \begin{cases} 0 & \text{if } v = \varepsilon \\ \varepsilon & \text{if } v = 0 \end{cases}$$

Max-plus linear systems

Max-plus linear system:

$$x(k) = A(k) \otimes x(k-1) \oplus B(k) \otimes u(k)$$

where

$k \in \mathbb{Z}$ = event counter.

$A \in \mathbb{R}_{\varepsilon}^{n \times n}$ = system matrix in cycle k.

$B \in \mathbb{R}_{\varepsilon}^{n \times p}$ = system matrix in cycle k.

Switching max-plus linear system:

The system can run in different modes $\ell(k) \in \{1, \ldots, n_{\mathrm{m}}\}$:

SMPL system:

$$x(k) = A(\ell(k), k) \otimes x(k-1) \oplus B(\ell(k), k) \otimes u(k)$$

Switching function:

$$\ell(k) = \phi_{\mathrm{s}}(x(k-1), \ell(k-1), u(k), v(k))$$

Implicit SMPL system:

$$x(k) = \Big(\bigoplus_{i=0}^{\bar{\mu}} A^{(i)}(\ell(k), k) \otimes x(k-i) \Big) \oplus B(\ell(k), k) \otimes u(k)$$

Dynamic graph (Murota)

Definition:

A dynamic graph

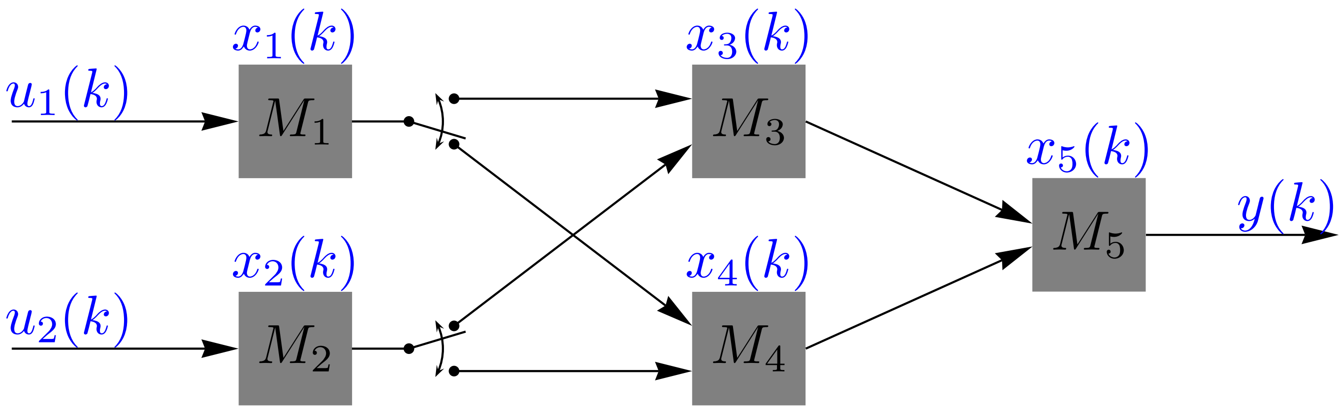
$$G = (G_0^1, \ldots, G_n^1, G_0^2, \ldots, G_n^2, \ldots, G_0^m, \ldots, G_n^m)$$

is a sequence of graphs, where $G_0^k = (X^k, E_0^k)$ is a directed graph with only nonpositive circuit weights, and $G_\mu^k = (X^k, X^{k-\mu}, E_\mu^k)$, $\mu = 1, \ldots, n$ is a directed bipartite graph where E_μ^k being the set of edges from $X^{k-\mu}$ to X^k. The nodes X_k represent the state of a system at event step k. The weight of the edge of G_0^k from node $[X^k]_j$ to $[X^k]_i$ is equal to $[A^{(0)}(\ell(k))]_{ij}$, The weight of the edge of G_μ^k from node $[X^{k-\mu}]_j$ to $[X^k]_i$ is equal to $[A^{(\mu)}(\ell(k))]_{ij}$.

Examples of SMPL systems

- production system
- printer
- legged robot
- container terminal
- railway network

production system



$$x_1(k) = \max(x_1(k-1) + d_1, u_1(k))$$

$$x_2(k) = \max(x_2(k-1) + d_2, u_2(k))$$

$$x_3(k) = \max(x_1(k) + d_1, x_3(k-1) + d_3)$$

$$x_4(k) = \max(x_2(k) + d_2, x_4(k-1) + d_4)$$

$$x_5(k) = \max(x_3(k) + d_3, x_4(k) + d_4, x_5(k-1) + d_5)$$

$$y(k) = x_5(k) + d_5$$

leading to the following matrices for the first mode:

$$A^{(0)}(1) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ d_1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & d_3 & d_4 & \varepsilon \end{bmatrix} , \quad B(1) = \begin{bmatrix} 0 & \varepsilon \\ \varepsilon & 0 \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}$$

$$A^{(1)}(1) = \begin{bmatrix} d_1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & d_3 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & d_4 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & d_5 \end{bmatrix}$$

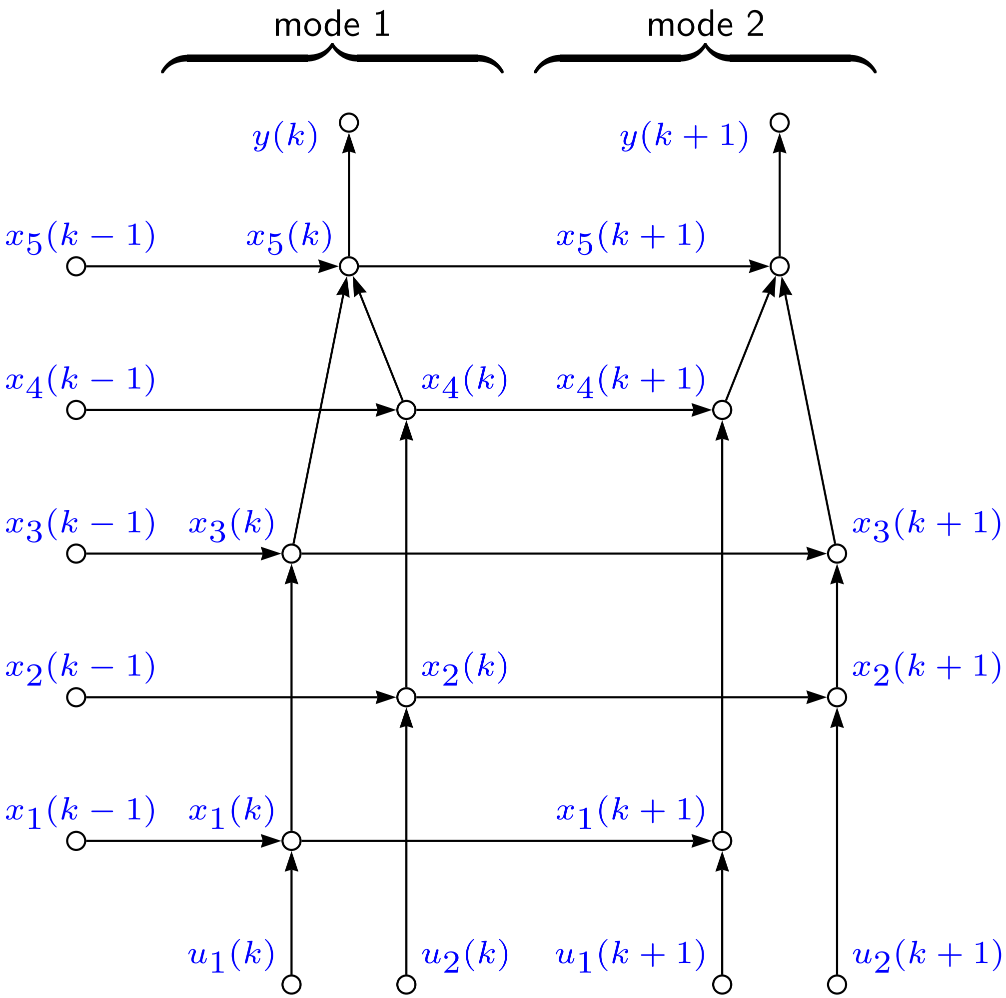
Second mode:

$$x_1(k) = \max(x_1(k-1) + d_1, u_1(k))$$
$$x_2(k) = \max(x_2(k-1) + d_2, u_2(k))$$
$$x_3(k) = \max(x_2(k) + d_2, x_3(k-1) + d_3)$$
$$x_4(k) = \max(x_1(k) + d_1, x_4(k-1) + d_4)$$
$$x_5(k) = \max(x_3(k) + d_3, x_4(k) + d_4, x_5(k-1) + d_5)$$
$$y(k) = x_5(k) + d_5$$

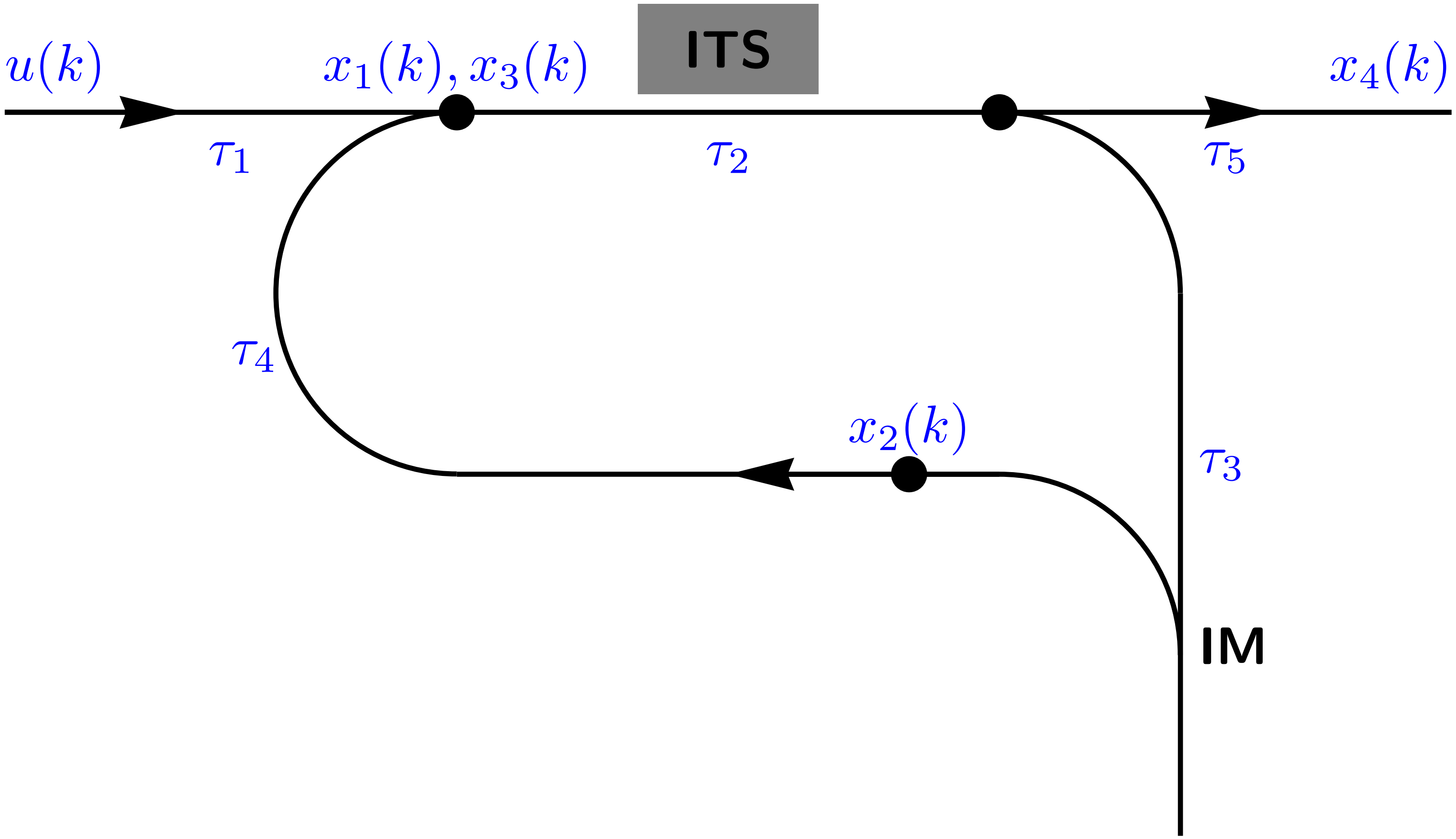
System matrices for the second mode:

$$A^{(0)}(2) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon & \varepsilon & \varepsilon \\ d_1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & d_3 & d_4 & \varepsilon \end{bmatrix}, \quad \begin{array}{c} A^{(1)}(2) = A^{(1)}(1) \\ B(2) = B(1) \end{array}$$

The dynamic graph for mode 1 in cycle k and mode 2 in cycle $k+1$:



Printer



Duplex printing:

$$x_1(k) = \max(u(k) + \tau_1, x_3(k-2) + \tau_2)$$

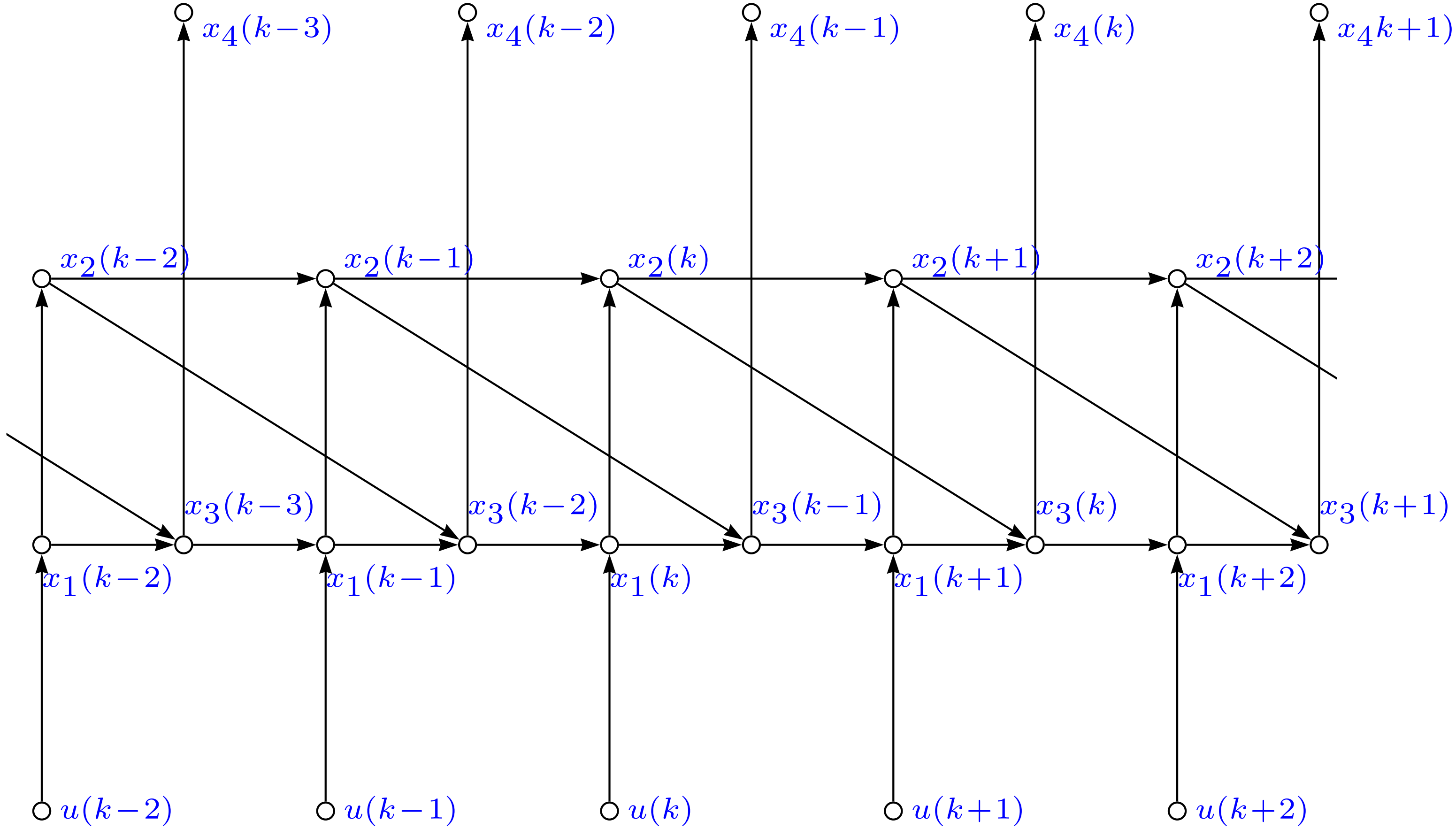
$$x_2(k) = \max(x_1(k) + \tau_2 + \tau_3, x_2(k-1) + \tau_2 + \tau_3)$$

$$x_3(k) = \max(x_1(k+1) + \tau_2, x_2(k) + \tau_4)$$

$$x_4(k) = x_3(k) + \tau_2 + \tau_5$$

$$A^{(0)}(1) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_2 + \tau_3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \tau_4 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \tau_2 + \tau_5 & \varepsilon \end{bmatrix}, \quad A^{(2)}(1) = \begin{bmatrix} \varepsilon & \varepsilon & \tau_2 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

$$A^{(1)}(1) = \begin{bmatrix} \varepsilon & \varepsilon & \tau_2 & \varepsilon \\ \varepsilon & \tau_2 + \tau_3 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \quad A^{(-1)}(1) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \quad B(1) = \begin{bmatrix} \tau_1 \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix}$$


$x_4(k-3)$
$x_4(k-2)$
$x_4(k-1)$
$x_4(k)$
$x_4 k+1)$
$x_2(k-2)$
$x_2(k-1)$
$x_2(k)$
$x_2(k+1)$
$x_2(k+2)$
$x_3(k-3)$
$x_3(k-2)$
$x_3(k-1)$
$x_3(k)$
$x_3(k+1)$
$x_1(k-2)$
$x_1(k-1)$
$x_1(k)$
$x_1(k+1)$
$x_1(k+2)$
$u(k-2)$
$u(k-1)$
$u(k)$
$u(k+1)$
$u(k+2)$

Simplex printing:

$$x_1(k) = x_3(k-2)$$

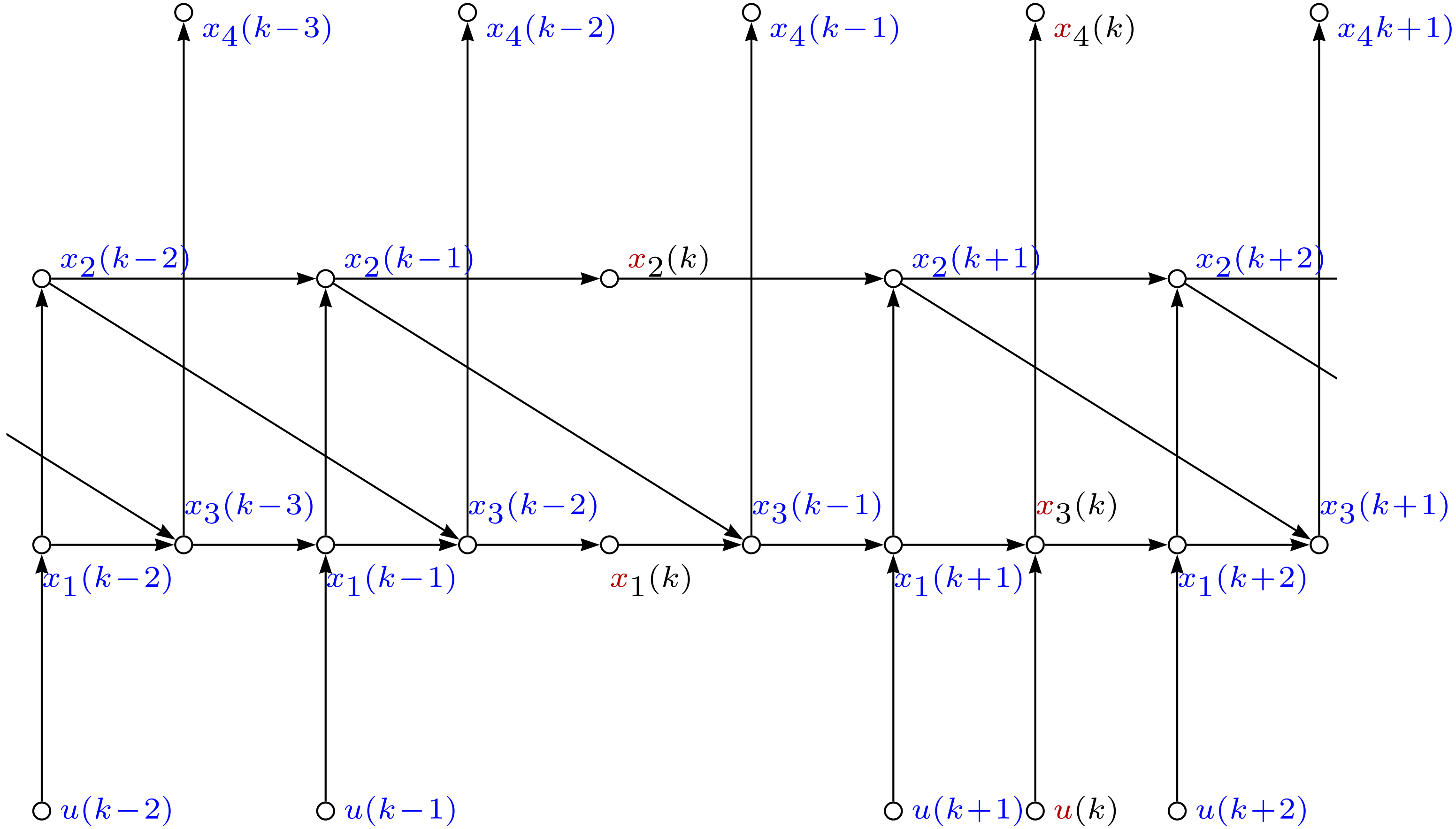
$$x_2(k) = x_2(k-1)$$

$$x_3(k) = \max(x_1(k+1) + \tau_2, u(k) + \tau_1)$$

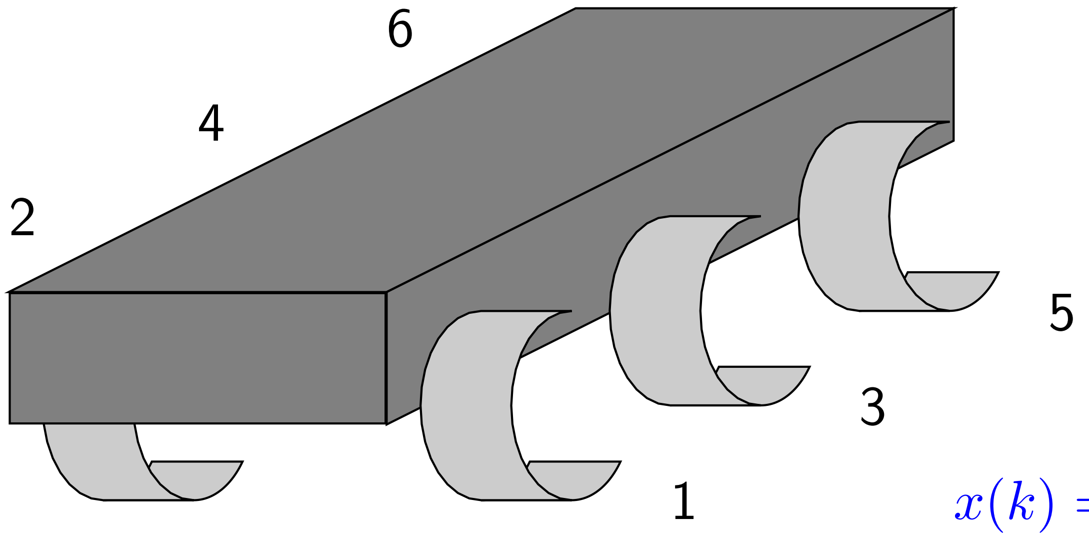
$$x_4(k) = x_3(k) + \tau_2 + \tau_5$$

$$A^{(0)}(1) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \tau_2 + \tau_5 & \varepsilon \end{bmatrix}, \quad A^{(2)}(1) = \begin{bmatrix} \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

$$A^{(1)}(1) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \quad A^{(-1)}(1) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \quad B(1) = \begin{bmatrix} \varepsilon \\ \varepsilon \\ \tau_1 \\ \varepsilon \end{bmatrix}$$


x4(k−3)
x4(k−2)
x4(k−1)
x4(k)
x4k+1)
x2(k−2)
x2(k−1)
x2(k)
x2(k+1)
x2(k+2)
x3(k−3)
x3(k−2)
x3(k−1)
x3(k)
x3(k+1)
x1(k−2)
x1(k−1)
x1(k)
x1(k+1)
x1(k+2)
u(k−2)
u(k−1)
u(k+1)
u(k)
u(k+2)

Legged robots



$$x(k) = \begin{bmatrix} t(k) \\ a(k) \end{bmatrix} = \begin{bmatrix} t_1(k) \\ \vdots \\ t_6(k) \\ a_1(k) \\ \vdots \\ a_6(k) \end{bmatrix}$$

Tripod gait $\mathcal{L}_1 = \{2, 3, 6\}$ and $\mathcal{L}_2 = \{1, 4, 5\}$.

Tetrapod gait: $\mathcal{L}_1 = \{1, 4\}$, $\mathcal{L}_2 = \{3, 6\}$ and $\mathcal{L}_3 = \{2, 5\}$.

$$x(k) = \left[\begin{array}{c|c} \mathcal{E} & \tau_f \otimes E \\ \hline P & \mathcal{E} \end{array}\right] \otimes x(k) \oplus \left[\begin{array}{c|c} E & \mathcal{E} \\ \hline \tau_g \otimes E \oplus Q & E \end{array}\right] \otimes x(k-1)$$

where

$$[P]_{pq} = \begin{cases} \tau_\Delta, & \forall j \in \{1, m-1\}; \forall p \in \mathcal{L}_{j+1}; \forall q \in \mathcal{L}_j \\ \varepsilon & \text{otherwise} \end{cases}$$

$$[Q]_{pq} = \begin{cases} \tau_\Delta, & \forall p \in \mathcal{L}_1; \forall q \in \mathcal{L}_m \\ \varepsilon & \text{otherwise} \end{cases}$$

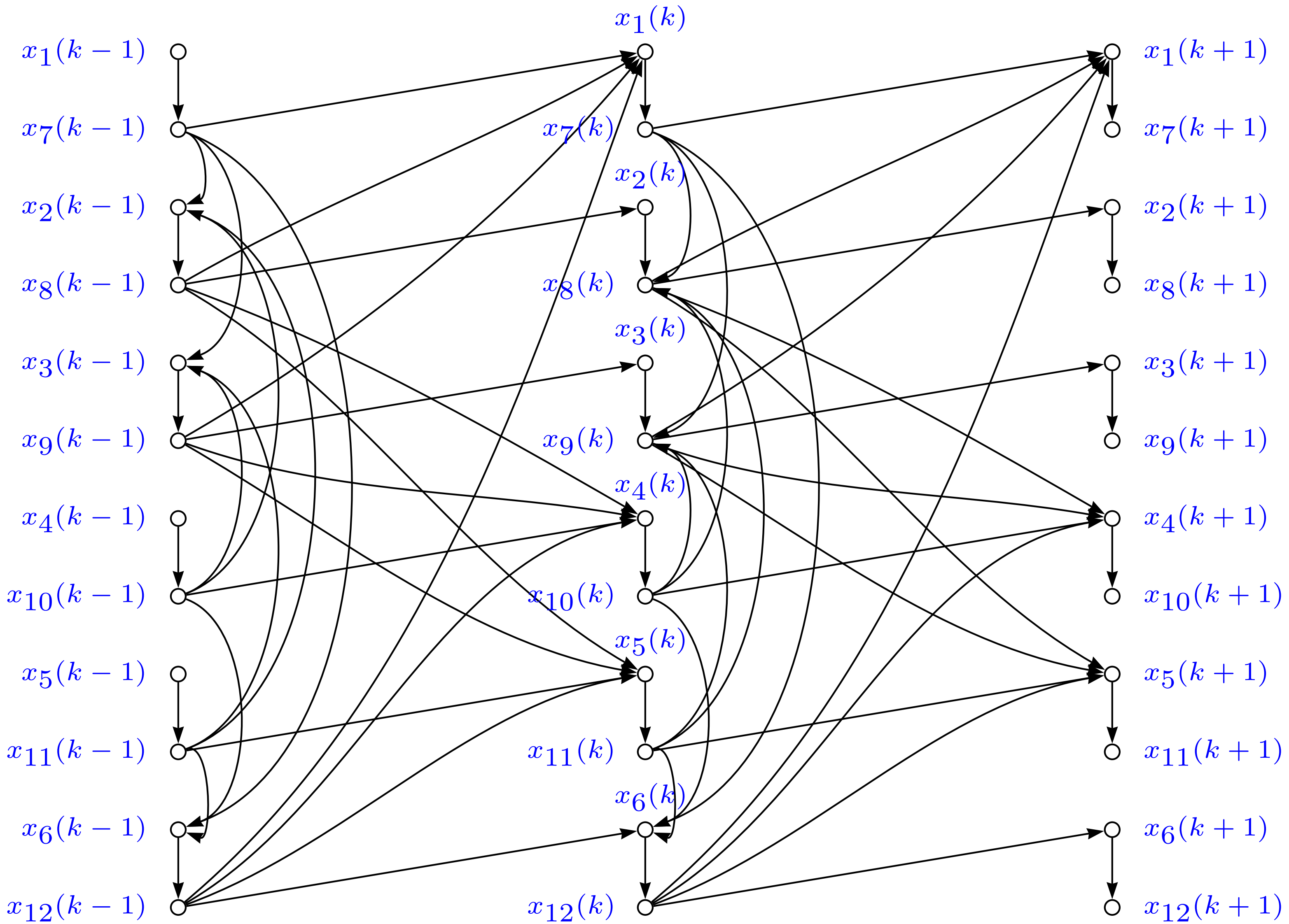
Tripod gait $\mathcal{L}_1 = \{2, 3, 6\}$ and $\mathcal{L}_2 = \{1, 4, 5\}$.
Tetrapod gait: $\mathcal{L}_1 = \{1, 4\}$, $\mathcal{L}_2 = \{3, 6\}$ and $\mathcal{L}_3 = \{2, 5\}$.

System matrices for tripod gait:

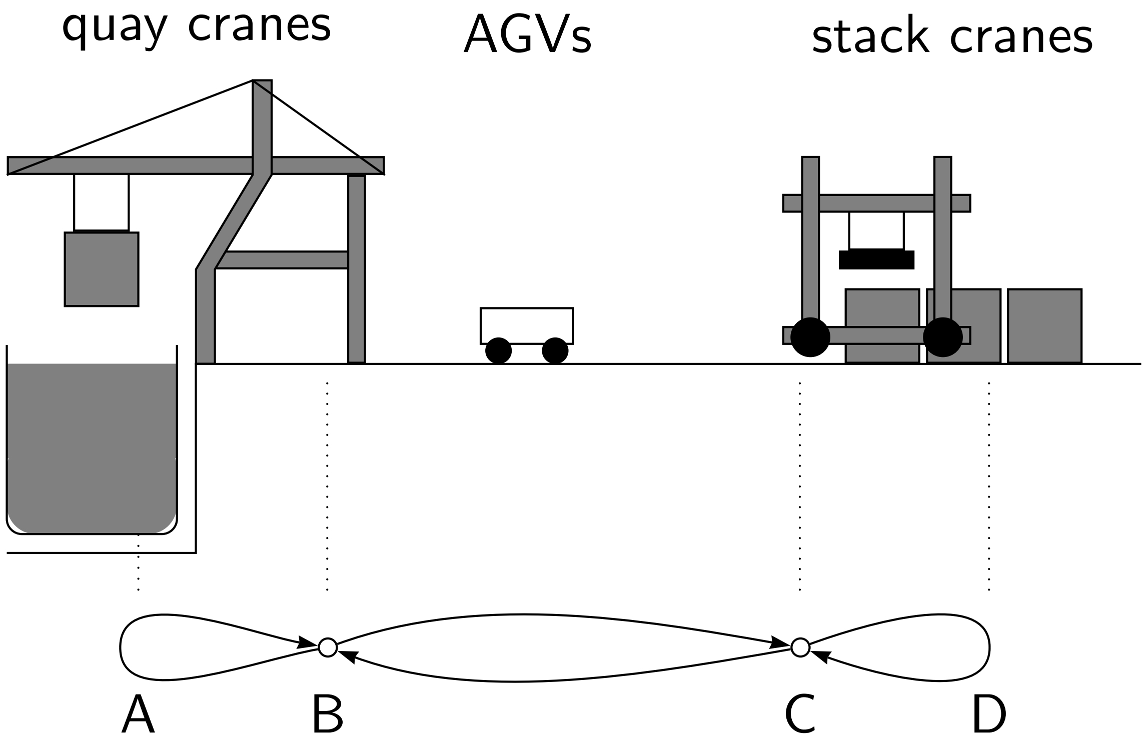
$$
A^{(0)}(1) = \left[\begin{array}{cccccc|cccccc}
\mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{T}_f & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} \\
\mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{T}_f & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} \\
\mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{T}_f & \mathcal{E} & \mathcal{E} & \mathcal{E} \\
\mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{T}_f & \mathcal{E} & \mathcal{E} \\
\mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{T}_f & \mathcal{E} \\
\mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{T}_f \\
\hline
\mathcal{E} & \mathcal{T}_\Delta & \mathcal{T}_\Delta & \mathcal{E} & \mathcal{E} & \mathcal{T}_\Delta & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} \\
\mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} \\
\mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} \\
\mathcal{E} & \mathcal{T}_\Delta & \mathcal{T}_\Delta & \mathcal{E} & \mathcal{E} & \mathcal{T}_\Delta & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} \\
\mathcal{E} & \mathcal{T}_\Delta & \mathcal{T}_\Delta & \mathcal{E} & \mathcal{E} & \mathcal{T}_\Delta & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} \\
\mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E}
\end{array}\right]
$$

and

$$A^{(1)}(1) = \left[\begin{array}{cccccc|cccccc}
0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\hline
\tau_g & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\tau_\Delta & \tau_g & \varepsilon & \tau_\Delta & \tau_\Delta & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\tau_\Delta & \varepsilon & \tau_g & \tau_\Delta & \tau_\Delta & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \tau_g & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \tau_g & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\
\tau_\Delta & \varepsilon & \varepsilon & \tau_\Delta & \tau_\Delta & \tau_g & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0
\end{array}\right]$$


$x_1(k-1)$
$x_7(k-1)$
$x_2(k-1)$
$x_8(k-1)$
$x_3(k-1)$
$x_9(k-1)$
$x_4(k-1)$
$x_{10}(k-1)$
$x_5(k-1)$
$x_{11}(k-1)$
$x_6(k-1)$
$x_{12}(k-1)$
$x_1(k)$
$x_7(k)$
$x_2(k)$
$x_8(k)$
$x_3(k)$
$x_9(k)$
$x_4(k)$
$x_{10}(k)$
$x_5(k)$
$x_{11}(k)$
$x_6(k)$
$x_{12}(k)$
$x_1(k+1)$
$x_7(k+1)$
$x_2(k+1)$
$x_8(k+1)$
$x_3(k+1)$
$x_9(k+1)$
$x_4(k+1)$
$x_{10}(k+1)$
$x_5(k+1)$
$x_{11}(k+1)$
$x_6(k+1)$
$x_{12}(k+1)$

Container terminal



k	$=$	be the number of the container,
$x_{\mathrm{q},i}(k)$	$=$	time of loading kth container on AGV at point B,
$x_{\mathrm{s},i}(k)$	$=$	time of unloading kth container from AGV at point C.

Define:

$$Q(k) = \text{quay crane that is handles container } k,$$
$$V(k) = \text{AGV that handles container } k,$$
$$S(k) = \text{stack crane that handles container } k.$$

$$\tau_{\mathrm{q}}(k) = \text{time quay crane needs to lift container from the ship},$$
$$\tau_{\mathrm{s}}(k) = \text{time stack crane needs to put container in the yard},$$
$$\tau_{\mathrm{u},j,i}(k) = \text{transp. time of unloaded vehicle from stack crane } j \text{ to quay crane } i,$$
$$\tau_{\mathrm{l},j,i}(k) = \text{transp. time of loaded vehicle from quay crane } i \text{ to stack crane } j.$$

Define the state

$$x(k) = \begin{bmatrix} x_{\mathrm{q}}^T(k) & x_{\mathrm{s}}^T(k) \end{bmatrix}^T$$
$$= \begin{bmatrix} x_{q,1}(k) & \cdots & x_{\mathrm{q},N_{\mathrm{q}}}(k) & x_{s,1}(k) & \cdots & x_{\mathrm{s},N_{\mathrm{s}}}(k) \end{bmatrix}^T$$

For $x_{\mathrm{q},i}(k)$ and $x_{\mathrm{s},j}(k)$ we derive

$$x_{\mathrm{q},i}(k) = \begin{cases} \max\Big(x_{\mathrm{q},i}(k-1) + \tau_{\mathrm{q}}(k), x_{\mathrm{s},j}(k-m(k)) + \tau_{\mathrm{u},j,i}(k)\Big) \\ \qquad\qquad \text{if } \ i = Q(k), j = S(k-m(k)) \\ x_{\mathrm{q},i}(k-1) \qquad \text{if } \ i \neq Q(k) \end{cases}$$

$$x_{\mathrm{s},j}(k) = \begin{cases} \max\left(x_{\mathrm{q},i}(k) + \tau_{\mathrm{l},i,j}(k), x_{\mathrm{s},j}(k-1)\right) \\ \qquad\qquad \text{if } \ j = S(k) \text{ and } i = Q(k) \\ x_{\mathrm{s},i}(k-1) \qquad \text{otherwise} \end{cases}$$

where

$$m(k) = \max_{\ell>0}\{\ell | V(k-\ell) = V(k)\},$$

State matrices A_i, $i = 0, 1, \ldots$:

$$[A^{(0)}]_{ij}(k) = \begin{cases} \tau_{1,i,j}(k) & \text{if } j = S(k) \text{ and } i = Q(k) \\ \varepsilon & \text{otherwise} \end{cases}$$

$$[A^{(1)}(k)]_{ij} = \begin{cases} \tau_{\mathrm{q}}(k) & \text{if } i = j, i = Q(k), i \leq N_{\mathrm{q}} \\ \tau_{\mathrm{q}}(k) & \text{if } i = j, i = N_{\mathrm{q}} + S(k), i > N_{\mathrm{q}} \\ 0 & \text{if } i = j, i \neq Q(k) \\ \tau_{\mathrm{u},j,i}(k) & \text{if } i \neq j, m(k) = 1, i = Q(k), \\ & \quad j = S(k-1) \\ \varepsilon & \text{otherwise} \end{cases}$$

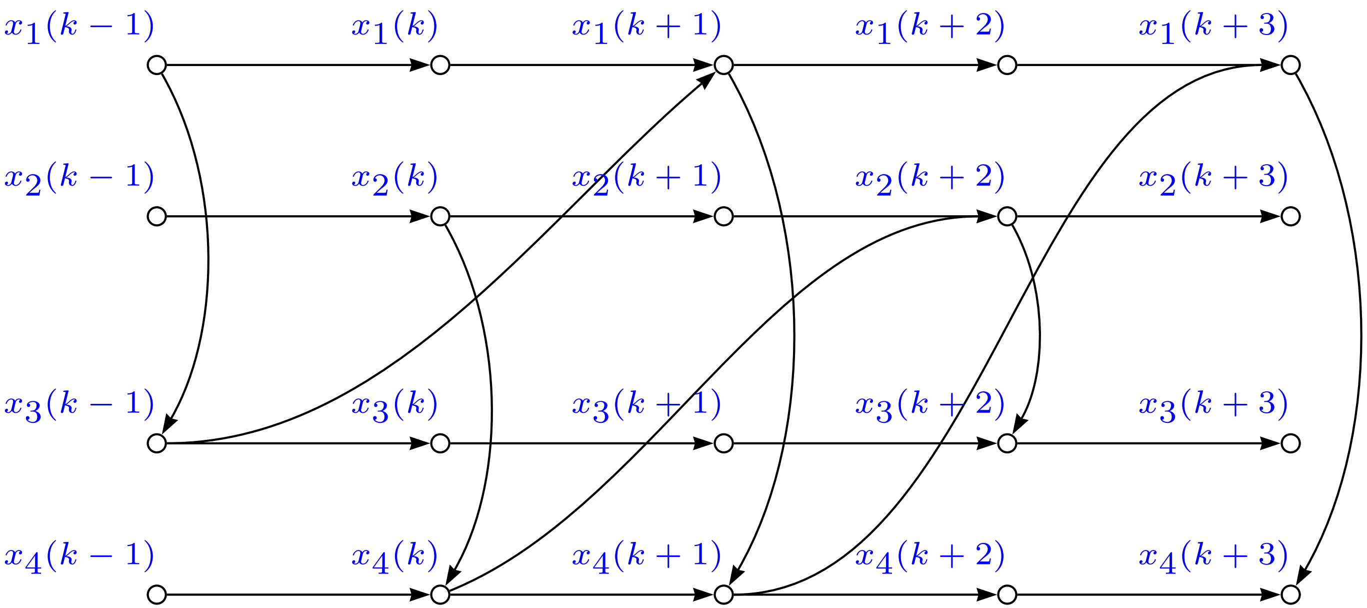
$$[A^{(\mu)}(k)]_{ij} = \begin{cases} \tau_{\mathrm{u},j,i}(k) & \text{if } i \neq j, m(k) = \mu, i = Q(k), \\ & \quad j = S(k-\mu) \\ \varepsilon & \text{otherwise} \end{cases}$$

Consider a small container terminal with $N_q = N_v = N_s = 2$.

$$\left[\; Q(k{-}1)\; Q(k)\; Q(k{+}1)\; Q(k{+}2) \;\right] = \left[\; 1 \;\; 2 \;\; 1 \;\; 2 \;\right]$$

$$\left[\; V(k{-}1)\; V(k)\; V(k{+}1)\; V(k{+}2) \;\right] = \left[\; 1 \;\; 2 \;\; 1 \;\; 2 \;\right]$$

$$\left[\; S(k{-}1)\; S(k)\; S(k{+}1)\; S(k{+}2) \;\right] = \left[\; 1 \;\; 2 \;\; 2 \;\; 1 \;\right]$$



Scheduling with SMPL systems

Semi-cyclic discrete event systems

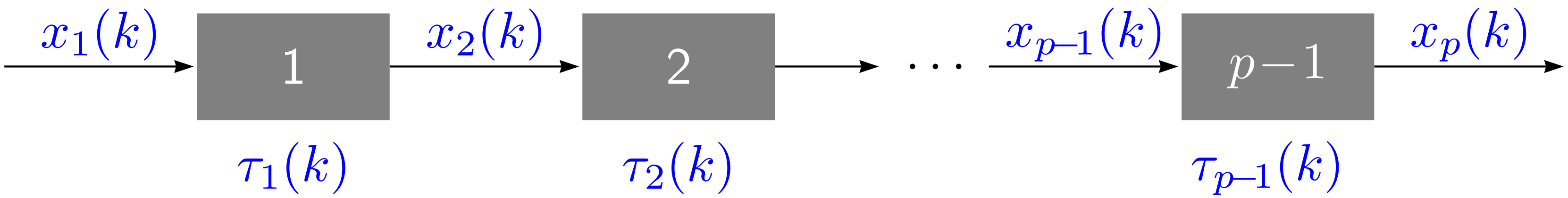
M jobs	L alternative routes
n operations	N resources

Derive Switching Max-Plus-Linear model with 3 basic types of decisions:

- Routing
- Ordering
- Synchronization

Routing

Job with $(p-1)$ operations



Routing equations:

$$x_2(k) \geq x_1(k) + \tau_1(k)$$

$$x_3(k) \geq x_2(k) + \tau_2(k)$$

$$\vdots$$

$$x_p(k) \geq x_{p-1}(k) + \tau_{p-1}(k)$$

In max-plus matrix notation this can be written as

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_p(k) \end{bmatrix} \geq \begin{bmatrix} \varepsilon & \varepsilon & \dots & \varepsilon \\ \tau_1(k) & \varepsilon & & \varepsilon \\ \vdots & \ddots & \ddots & \vdots \\ \varepsilon & \dots & \tau_{p-1}(k) & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_p(k) \end{bmatrix}$$

or in short notation

$$x(k) \geq A_{\text{job}}(k) \otimes x(k)$$

Multiple cycles state equation:

$$A_{\text{job}}(k) = \bigoplus_{\mu=1}^{L} w_\mu \otimes A_{\text{job}}^{(\mu)}(k) \otimes x(k-\mu)$$

L alternative routes $\longrightarrow$ L different system matrices:

$$A^{(\mu)}_{\mathrm{job},\ell}(k) \quad \text{for} \quad \ell = 1, \ldots, L$$

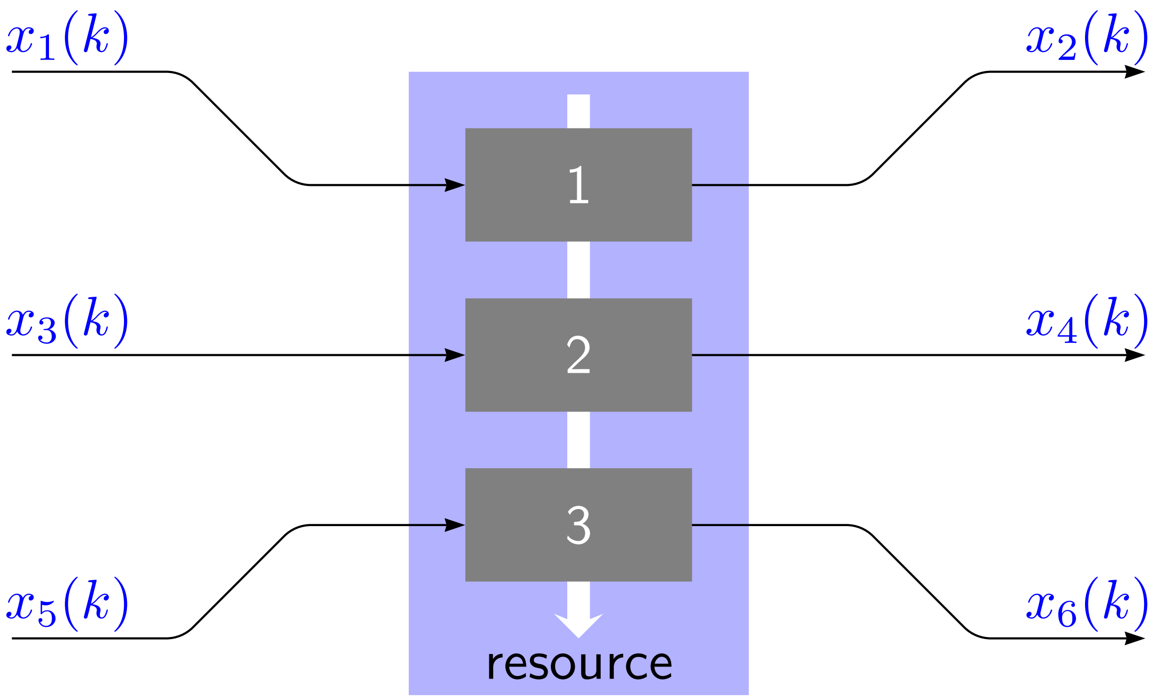
Max-plus binary variables $\omega_i(k)$, $i = 1, \ldots, L$ such that for route ℓ we have

$$\omega_\ell(k) = 0 \quad \text{and} \quad \omega_i(k) = \varepsilon \quad \text{for} \quad i \neq \ell$$

Job-system matrices:

$$A^{(\mu)}_{\mathrm{job}}(\omega(k), k) = \bigoplus_{\ell=1}^{L} \omega_\ell(k) \otimes A^{(\mu)}_{\mathrm{job},\ell}(k)$$

Ordering operations on resources



n operations

N resources

L alternative routes

$\omega_\ell(k)$ routing variables

$$[P_\ell]_{ij} = \begin{cases} 0 & \text{if operation } i \text{ and operation } j \text{ are on same resource} \\ \varepsilon & \text{if operation } i \text{ and operation } j \text{ are \underline{not} on same resource} \end{cases}$$

Matrix $P(\omega(k))$ for selection of the resources:

$$P(\omega(k)) = \bigoplus_{\ell=1}^{L} \omega(k) \otimes P_\ell$$

Separation matrix

$$[H]_{i,j}(k) = \begin{cases} \tau^{\mathrm{o}}_{i,j}(k) & \text{if operations } i \text{ and } j \text{ may be on same resource} \\ \varepsilon & \text{if operations } i \text{ and } j \text{ are } \underline{\text{never}} \text{ on same resource} \end{cases}$$

Order max-plus binary decision matrices

$$[Z^{(\mu)}]_{i,j}(k) = \\ \begin{cases} 0 & \text{if operation } i \text{ in cycle } k \text{ is after operation } j \text{ in cycle } k+\mu \\ \varepsilon & \text{if operation } i \text{ in cycle } k \text{ is before operation } j \text{ in cycle } k+\mu \end{cases}$$

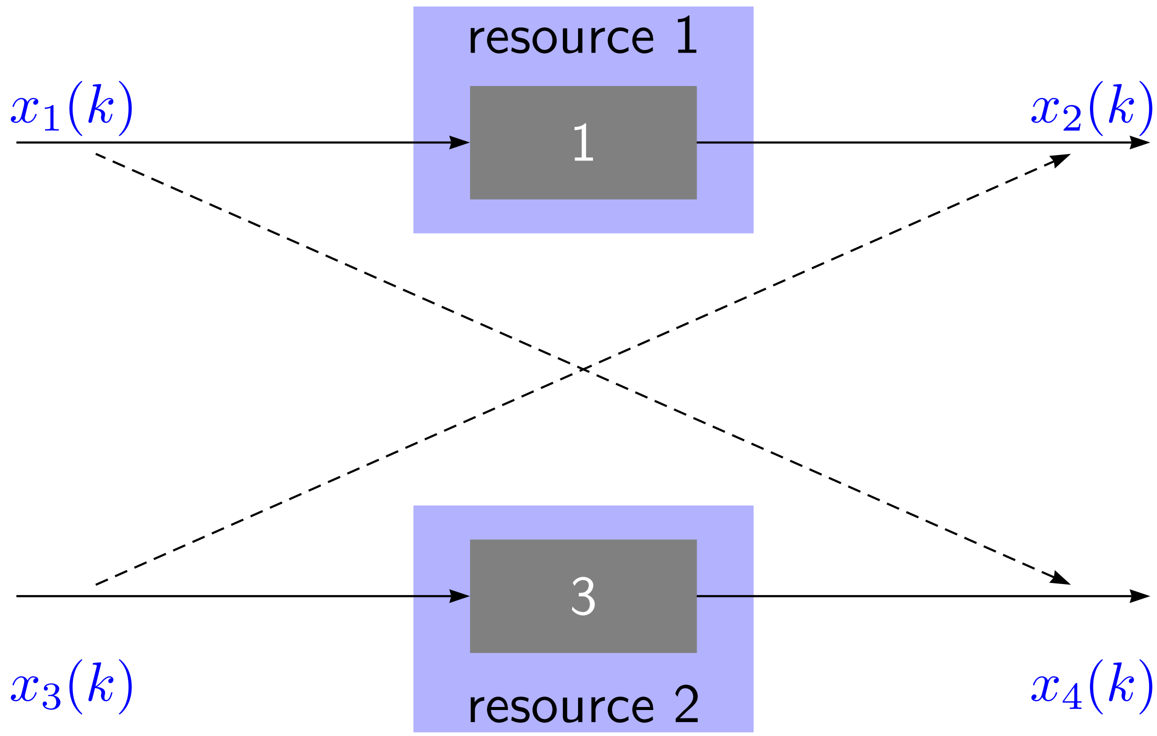
Ordering system matrices

$$A^{(\mu)}_{\mathrm{ord}}(\omega(k), \gamma^{(\mu)}(k), k) = P(\omega(k)) \odot Z^{((\mu))}(\gamma^{\mu}(k)) \odot H(k)$$

Ordering constraints in the system

$$x(k) \geq \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A^{(\mu)}_{\mathrm{ord}}(\omega(k), \gamma^{(\mu)}(k), k) \otimes x(k-\mu)$$

Synchronization of operations



Synchronization between operations in different jobs, e.g.

- synchronization of legs in a legged robot.
- two trains on platform give passengers opportunity to change trains.

Define synchronization modes $\ell = 1, \ldots, L_{sync}$.

$$[A^{(\mu)}_{\mathrm{syn},\ell}(k)]_{ij} = \begin{cases} \tau^{\mathrm{s}}_{i,j}(k) & \text{if operation } j \text{ in cycle } k \text{ may be scheduled after} \\ & \qquad \text{operation } i \text{ in cycle } k{+}\mu \\ \varepsilon & \text{elsewhere} \end{cases}$$

Define max-plus binary synchronization variable $s(k)$.

The synchronization system matrix is given by

$$A^{(\mu)}_{\mathrm{syn}}(\sigma(k), k) = \bigoplus_{\ell=0}^{L_{\mathrm{syn}}} \sigma_\ell(k) \otimes A^{(\mu)}_{\mathrm{syn},\ell}(k)$$

and the operation synchronization constraints become:

$$x(k) \geq \bigoplus_{\mu=1}^{\bar{\mu}} A^{(\mu)}_{\mathrm{syn}}(s(k), k) \otimes x(k - \mu).$$

Overall MPL system

Max-plus binary decision variables

- Routing: $w(k)$
- Ordering: $\gamma^{(\mu)}(k)$
- Synchronization: $s(k)$

Stack all decision variables into one vector

$$v(k) = \begin{bmatrix} w(k) \\ \gamma^{(0)}(k) \\ \vdots \\ \gamma^{(\bar{\mu})}(k) \\ s(k) \end{bmatrix} \in (\mathbb{B}_\varepsilon)^{L_{\mathrm{tot}}}$$

where L_{tot} is the total number of scheduling variables.

Define overall system matrix

$$A^{(\mu)}(v(k),k) = A_{\text{job}}^{(\mu)}(\omega(k),k) \oplus A_{\text{ord}}^{(\mu)}(\omega(k),\gamma^{(\mu)}(k),k) \oplus A_{\text{syn}}^{(\mu)}(\sigma(k))$$
$$= \bigoplus_{\ell=1}^{L_{\text{tot}}} v_\ell(k) \otimes A_{\text{tot},\ell}^{(\mu)}(k)$$

Matrix $A^{(\mu)}$ is max-plus affine in the control variables $v(k)$.
The scheduling model is as follows

$$x(k) = \bigoplus_{\mu=0}^{\bar{\mu}} A^{(\mu)}(v(k),k) \otimes x(k-\mu) \oplus r(k)$$

Control vector $v(k)$ decides on mode of operation.

Model Predictive Scheduling

Receding horizon principle

- Not schedule for the complete task
- In several iterations with prediction horizon (only jobs in nearest future)

Model Predictive Scheduling problem at time t:

$$\min_{v(k+j,t),\ j=0,\ldots,N_\mathrm{p}-1} J(k,t)$$

subject to

$$x(k+j,k+j,t) = \bigoplus_{\mu=0}^{\bar{\mu}} A^{(\mu)}(v(k+j,t),k+j,t) \otimes x(k+j-\mu,t) \oplus r(k+j)$$

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

$$J(k,t) = \delta \max_{i=1,\ldots,n} x_i(k+N_\mathrm{p},t) + \sum_{j=0}^{N_\mathrm{p}-1} \sum_{i=1}^{n} \kappa_{j,i}\, x_i(k+j,t)$$
$$+ \sum_{j=0}^{N_\mathrm{p}-1} \sum_{i=1}^{n} \lambda_i \max\Big(x_i(k+j,t) - x_{\mathrm{d},i}(k+j)\;,\; 0 \Big)$$
$$- \sum_{j=0}^{N_\mathrm{p}-1} \sum_{m=1}^{n_u} \rho_{j,m}\, u_m(k+j,t) + \sum_{l=1}^{L_\mathrm{tot}} \sigma_{j,l}\, v_l^\flat(k+j,t).$$

where

$$v_l^\flat(k+j,t) = \begin{cases} 0 & \text{for } v_l(k+j,t) = \varepsilon \\ 1 & \text{for } v_l(k+j,t) = 0 \end{cases}$$

is a conventional binary variable.

Mixed-Integer Linear Programming

The model predictive scheduling problem can be recast into a mixed-integer linear programming problem as follows:

- Use the following approximation

$$v_i(k,t) = \beta\,(1 - v_i^\flat(k,t))$$

  where $\beta \ll 0$ is a very large (in absolute value) negative number.

- Max-plus constraints become linear constraints.

- Object function becomes linear function.

There exist fast and reliable solvers (e.g. CPLEX, Xpres) for MILP.

Part II: Application on Railway traffic management

- Dutch railway network
- Minimize sum of delays
- Disturbances: small perturbations - handled by reordering trains.
- Disruptions: blocked tracks lead to large decrease in network capacity.
- Develop decision support systems for the dispatchers.
- Model predictive scheduling approach.
- Centralized MPS $\longrightarrow$ Distributed MPS.
- Macroscopic model with some specific microscopic features.

Railway traffic model

A max-plus linear model is used to predict the effects of the dispatching actions.

- local management of routing in stations and interlocking area.
- station and interlocking area are modeled as single point.
- track between points modeled as single segment.
- block sections/signaling not modeled explicitly.
- time separation $\rightarrow$ headway constraints.

State: $$x(k) = \begin{bmatrix} d_1(k) \\ \vdots \\ d_n(k) \\ a_1(k) \\ \vdots \\ a_n(k) \end{bmatrix}$$

Max-plus linear

Running time constraint models a train traversing a track.
Continuity constraint models a train dwelling at a station.
Headway constraints ensure a safe distance between trains on the same track.
Connection constraint models the transfers at stations.

The general form of these four constraints is:

$$x_i \geq x_j + \tau_{ij}$$

$x_i, x_j \in \mathbb{R}$ are departure and arrival times at stations. $\tau_{ij} \in \mathbb{R}$ is the minimum process time (dwell, running, headway, separation, or connection time).

Timetable constraints: For $r_i \in \mathbb{R}$ is the scheduled departure time

$$x_i \geq r_i$$

Switching max-plus linear model

For changing the order of the trains we adapt constraints with control variables

$$x_i \geq x_j + \tau_{ij} + (\gamma_{ij} + \omega_{ij}) \tag{1}$$

$$x_j \geq x_i + \tau_{ji} + (\bar{\gamma}_{ij} + \omega_{ij}), , \tag{2}$$

where γ_{ij} and ω_{ij} are max-plus binary control variables. Ordering variable γ_{ij} "enables" and "disables" constraints. Routing variable ω_{ij} decides if train i and j use the same track.

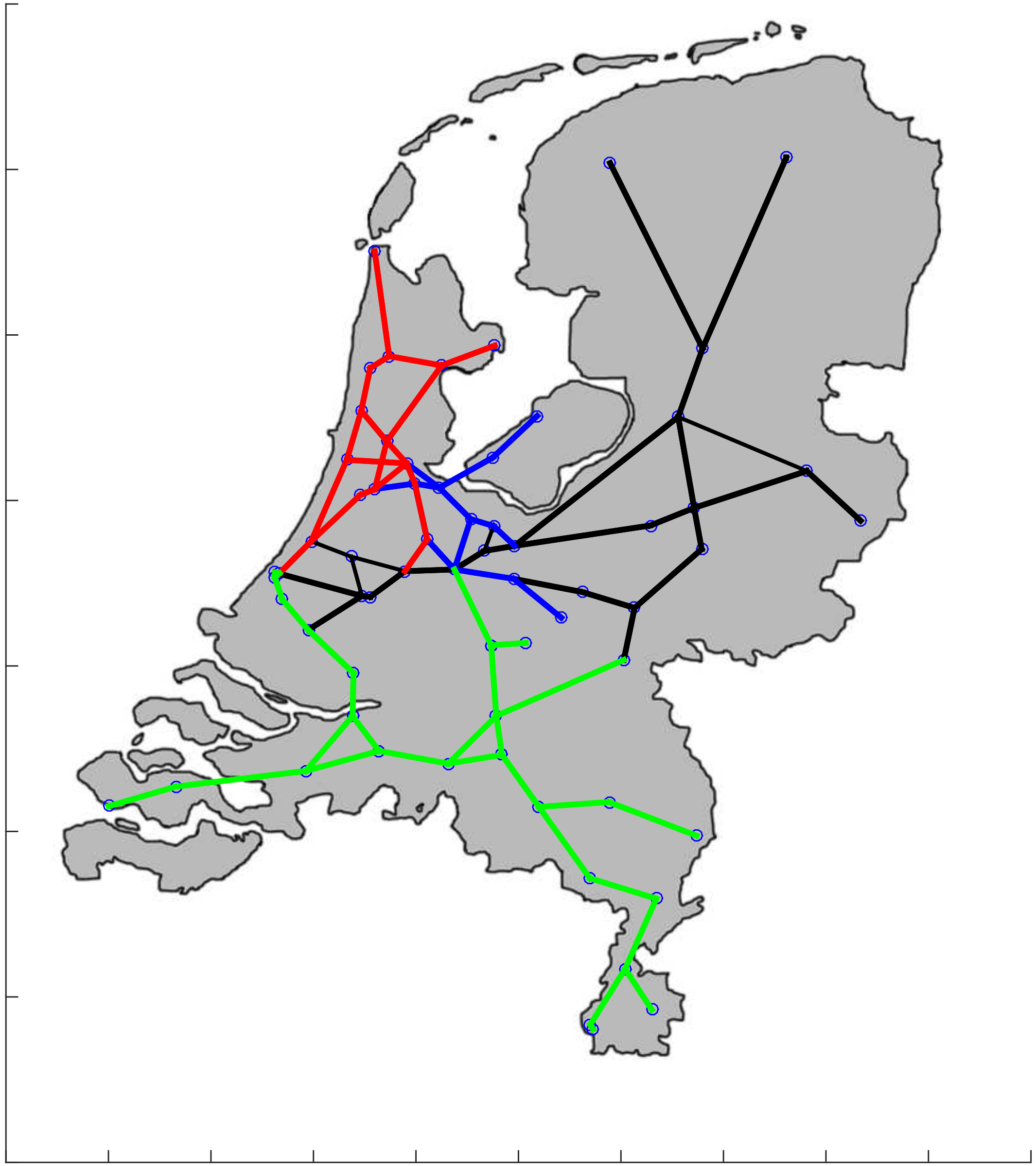
Model predictive scheduling of Dutch railway network

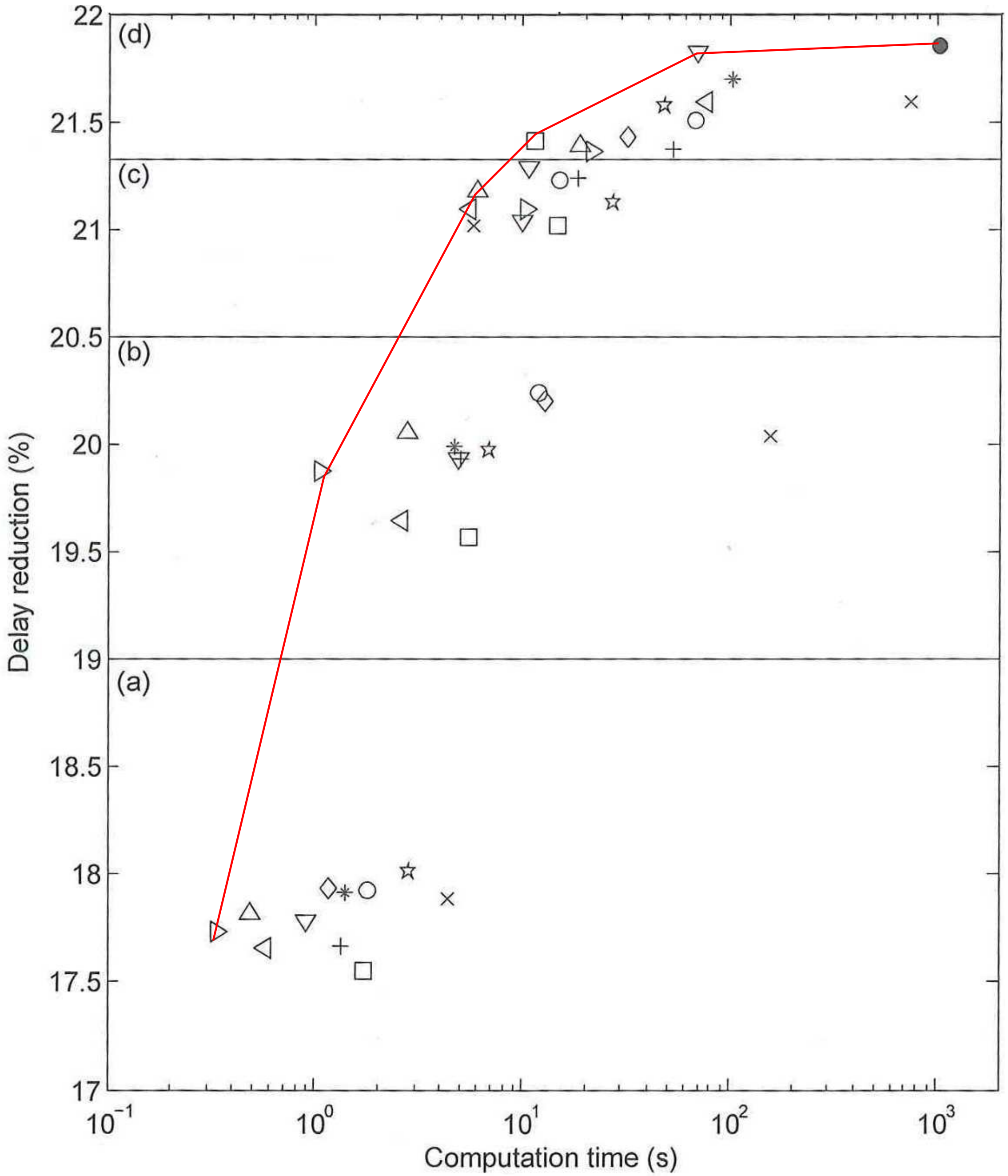
Dutch railway network:

- 326 train runs
- 1930 continuous variables
- 2744 binary variables
- 22050 constraints

Distributed Model Predictive Scheduling

- MILP is split up into several interacting smaller MILPs
- Each smaller MILP is solved separately
- Coordination between smaller MILPs to reach good traffic control
- For $N_p = 75$ minutes: computation time is less than 60 seconds.





$N_p = 75$ min

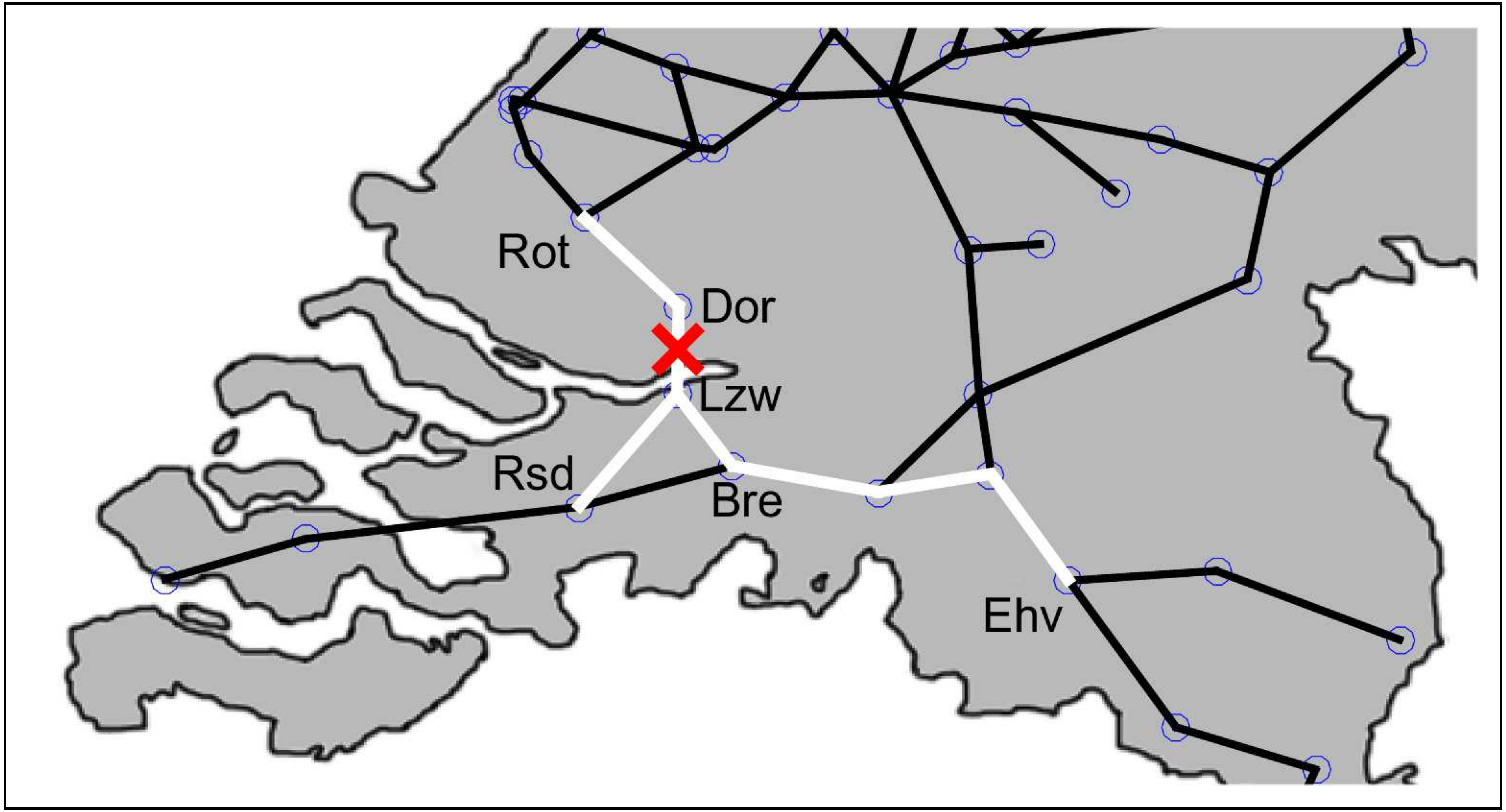
$N_p = 60$ min

$N_p = 45$ min

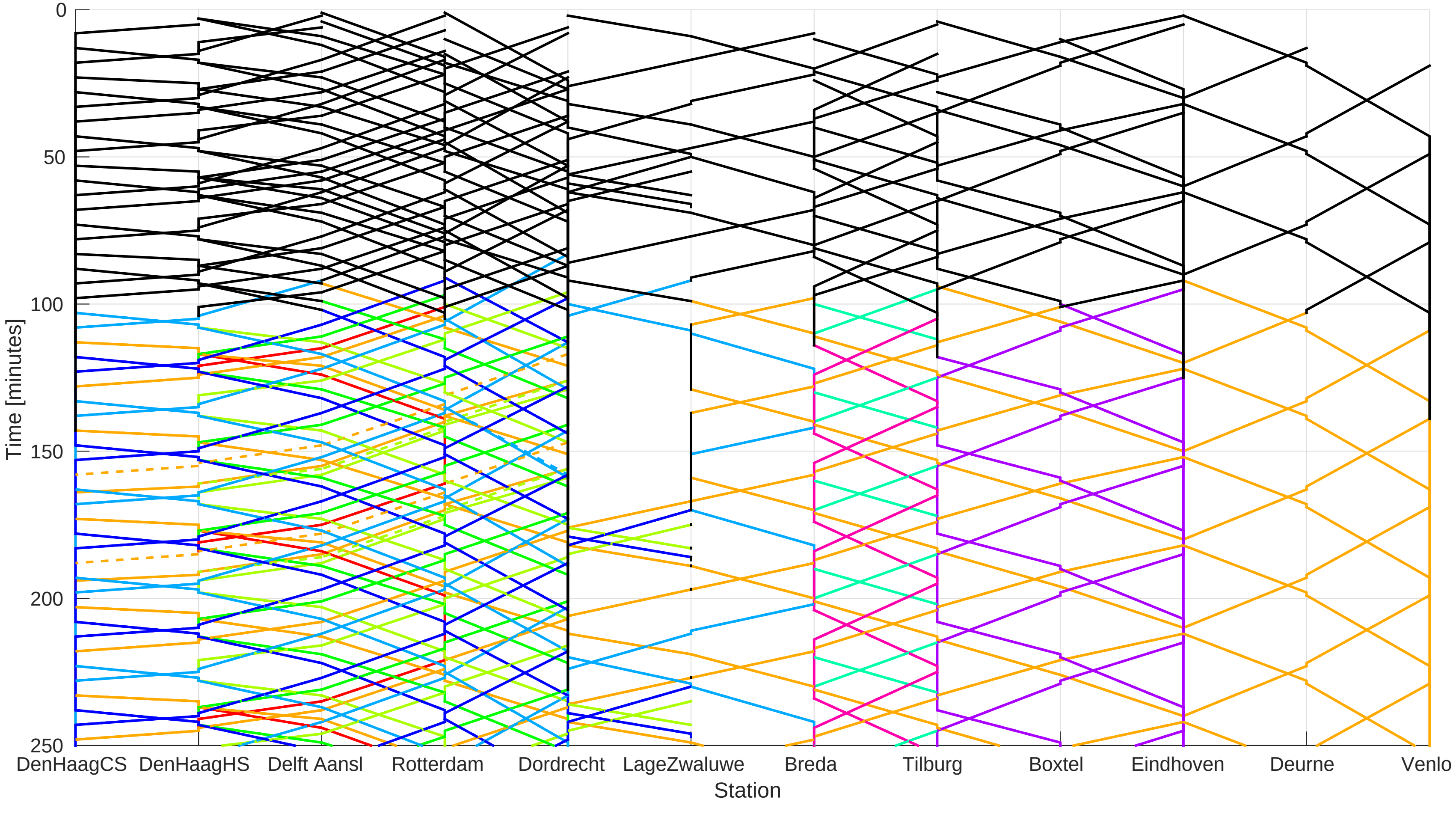
$N_p = 30$ min

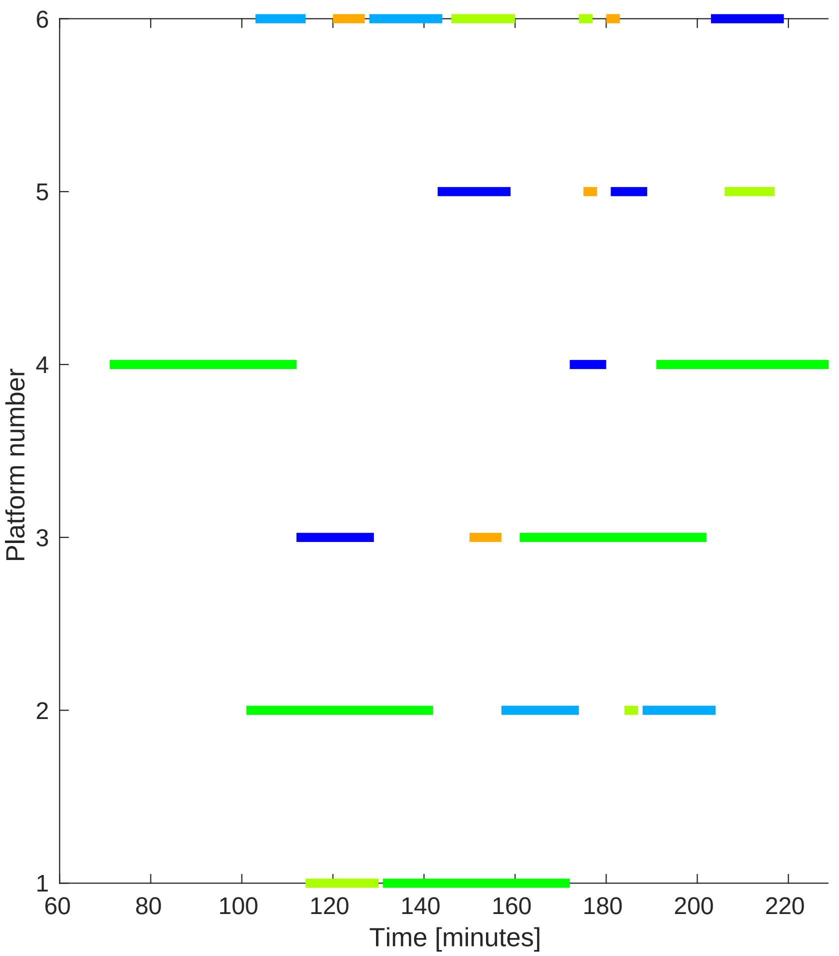
Disruption management

- Define disrupted region
- Cancelling trains
- Short-turn trains
- Shunting trains
- Platform assignment
- Distributed model predictive scheduling of disrupted network

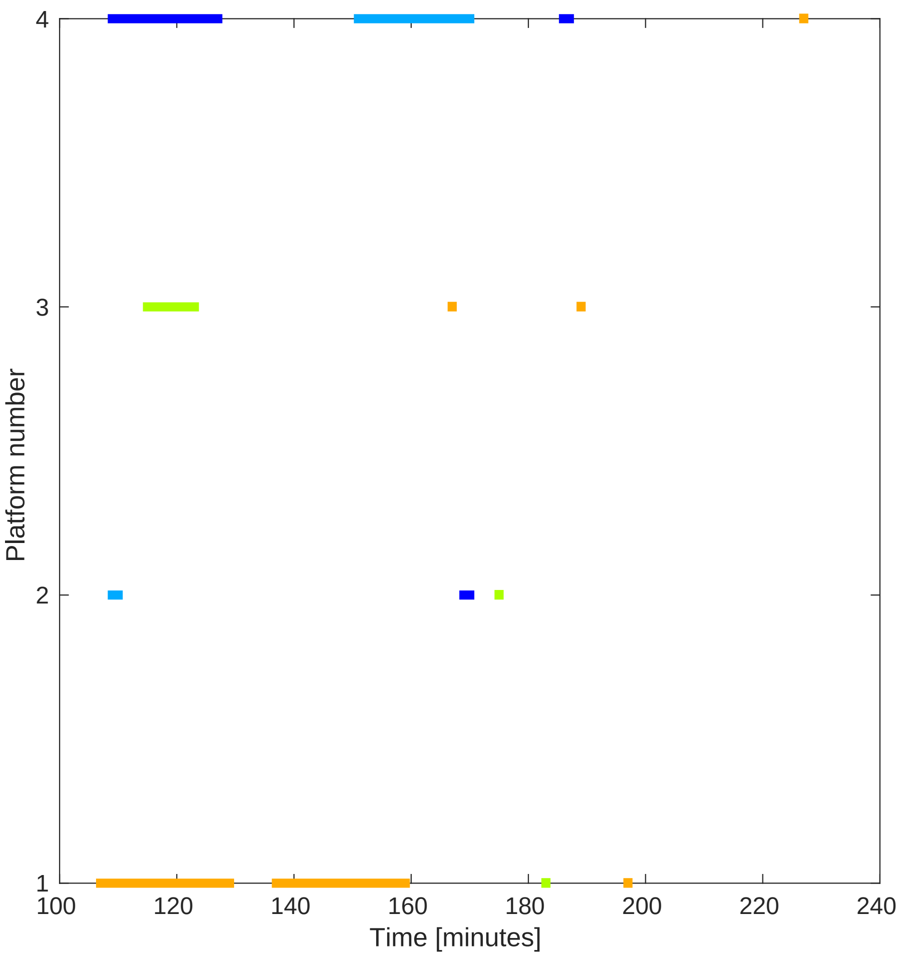


Rot
Dor
Lzw
Rsd
Bre
Ehv


0
50
100
150
200
250
Time [minutes]
DenHaagCS
DenHaagHS
Delft Aansl
Rotterdam
Dordrecht
LageZwaluwe
Breda
Tilburg
Boxtel
Eindhoven
Deurne
Venlo
Station



A. Station Dordrecht



B. Station Lage-Zwaluwe

Discussion

- semi-cyclic discrete-event systems can be modeled as SMPL systems
- switching max-plus linear systems in the context scheduling
- scheduling problem can be recast as a mixed-integer linear program
- rescheduling of the Dutch disturbed railway network
- rescheduling of the Dutch disrupted railway network
  1. Cancelling trains
  2. Short-turn trains
  3. Shunting trains
  4. Platform assignment
- future work:
  1. study the effect of noise in SMPL systems
  2. reduce computation effort