Stability Analysis of Periodic Railway Timetables

A Max-Plus Algebra Approach

TRAIL Research School, Delft, November 2002

Author
Drs. Rob M.P. Goverde
Faculty of Civil Engineering and Geosciences, Delft University of Technology, Transportation Planning and Traffic Engineering Section

© 2002 by Drs. Rob M.P. Goverde and TRAIL Research School
1.1 Background

Train operations are typically exposed to random variations in train running times and dwell times resulting in primary delays. Moreover, once a train is delayed this may produce severe delay propagation over the network. The sensitivity to delays of a train network timetable is managed by incorporating recovery times or buffer times. A stable timetable is able to deal with a certain amount of delay without intervention of process operators. Timetable robustness refers to the amount of effort for schedule adherence after disruptions. Evaluating a train network timetable on robustness is an important part of the timetable design process and typically requires a computer-aided approach, as the (circular) interactions of train movements on a railway network are very complex.

Performance of a rail traffic network depends on many aspects. However, the operational traffic conditions are completely summarized in the timetable. A timetable implicitly includes the set of train lines, train characteristics, and rail infrastructure (railway network and signalling system) Alternative line systems or altered allocations of train types to lines (and thereby changed running times) result in distinct network timetables that can be evaluated separately.

Methods capable of analysing large-scale highly-interconnected timetables are rare. Simulation is usually pursued to analyse complex systems when no analytical tools are available. However, simulation is typically very time-consuming, which forces railway planners to concentrate on stations or rail corridors and thereby discarding network interdependencies. Recently some tools have been developed that analyse train timetables on large-scale railway networks. FASTA (Nordeen, 1996) is a Swiss stability analysis tool based on simulation. In the Netherlands, the simulation system SIMONE (Middelkoop and Bouwman, 2002) has been developed for testing stability. This tool is compatible with the Dutch timetable design system DONS (Hooghiemstra, 1994). CAPRES (Lucchini et al., 2000) is a recent developed Swiss analytical capacity analysis system based on saturation of timetable variants.

This paper presents an analytical stability analysis method based on max-plus algebra. The applicability of max-plus algebra modelling and analysis of railway systems has first been investigated by Braker (1993), who derived a max-plus model of the Dutch intercity network. Subiono (2000) extended this approach to the modelling of the total Dutch train network including different train types (regional, interregional, and intercity). Goverde (2001,2002) added headway constraints to the max-plus modelling reflecting infrastructure restrictions. An overview of the max-plus theory applied to railway systems and efficient numerical algorithms for large-scale networks is given in Goverde and Soto y Koelmeijer (2000).

At the end of the 1990s it was decided to implement the max-plus modelling and analysis methods into a software application to automate the extensive modelling procedures and make the max-plus analysis approach accessible to railway planners. A first prototype established the utilization of DONS database files to automatically
\[ x_{1901, \text{Ehv}}(k) \geq 22 + x_{801, \text{Ht}}(k-1). \]

The scheduled departure time at Eindhoven \( d_{1901, \text{Ehv}}(k) \) is 1 minute after the whole hour, yielding the timetable constraint

\[ x_{1901, \text{Ehv}}(k) \geq d_{1901, \text{Ehv}}(k), \]

where \( d_{1901, \text{Ehv}}(k) = 01 + (k-1) \cdot 60. \)

Recall, that the first departure is at 01, the second at 61, then at 121, et cetera. Finally, a departure headway at Eindhoven of at least 3 minutes has to be respected with the local 6500 train line to Venlo, that is scheduled at 33 each hour. This infrastructure constraint is modelled by two inequalities since the headway constraint holds for both the previous and next local train of the 6500 line, i.e.,

\[ x_{1901, \text{Ehv}}(k) \geq 3 + x_{6501, \text{Ehv}}(k-1) \]
\[ x_{6501, \text{Ehv}}(k) \geq 3 + x_{1901, \text{Ehv}}(k). \]

In the same way headway constraints are obtained for any possible conflict between train arrivals and/or departures. If we collect all constraints we obtain the following (max,+) recursion

\[ x_{1901, \text{Ehv}}(k) = \max(26 + x_{1901, \text{Tb}}(k-1), 22 + x_{801, \text{Ht}}(k-1), 3 + x_{6501, \text{Ehv}}(k-1), d_{1901, \text{Ehv}}(k)). \]

Note that the second headway constraint will be present in the recursion for the departure of the 6500 train to Venlo. In this way we can model all interactions between trains in a train network. The resulting model is a (max,+) recursion, which is represented more elegantly in the so-called max-plus algebra. This is shown in the next section.

### 2.2 Max-Plus Linear Systems

We introduce a new notation for the two operations maximum and + as follows:

\[ a \oplus b := \max(a, b) \]
\[ a \oplus b := a + b, \]

where \( a \) and \( b \) are two real scalars and including the element \(-\infty\), for which the following calculation rules hold: \( a \oplus -\infty = a \) and \( a \oplus -\infty = -\infty \) for all real scalars \( a \).

Using this notation the recursive equation derived in the former section can be written as

\[ x_{1901, \text{Ehv}}(k) = (26 \otimes x_{1901, \text{Tb}}(k-1)) \oplus (22 \otimes x_{801, \text{Ht}}(k-1)) \oplus (3 \otimes x_{6501, \text{Ehv}}(k-1)) \oplus d_{1901, \text{Ehv}}(k). \]

In this formulation the recursion is clearly a linear equation although non-standard.
max-plus algebra. In particular also matrix computations can be introduced similarly to conventional linear algebra:

\[
(A \otimes B)_{ij} := (a_{ij} \otimes b_{ij}) \\
(A \otimes B)_{ij} := (a_{i1} \otimes b_{1j}) \oplus \ldots \oplus (a_{in} \otimes b_{nj}) = \bigoplus_{k=1}^{n} (a_{ik} \otimes b_{kj}).
\]

Using this matrix notation, the recursive equation is written as (check!)

\[
x_{1901,\text{Ehv}}(k) = \begin{bmatrix} 26 & 22 & 3 \end{bmatrix} \otimes \begin{bmatrix} x_{1901,\text{Tb}}(k-1) \\ x_{801,\text{Ehv}}(k-1) \oplus d_{1901,\text{Ehv}}(k) \\ x_{6501,\text{Ehv}}(k-1) \end{bmatrix}.
\]

In general, we obtain a max-plus linear system

\[
x(k) = A \otimes x(k-1) \oplus d(k),
\]

where the state vector \( x(k) \) contains all departure times in one vector, analogous for \( d(k) \), and the state matrix \( A \) is defined such that \( a_{ij} \) is the running time plus dwell/transfer/layover time or headway time (whatever is appropriate) if an interaction exists from \( x_j(k-1) \) to \( x_i(k) \), and \( a_{ij} = -\infty \) otherwise. Note that in the latter case the component \( a_{ij} \otimes x_j(k-1) = -\infty \otimes x_j(k-1) = -\infty \) does not contribute to the maximization, and thus has no effect on the departure time \( x_i(k) \).

Equation (1) is a first-order representation. Interactions with trains that departed more than one hour ago may also occur when running times are larger than 60 minutes. In this case also elements \( x(k-2) \) or even larger periods back are included in the linear system. Also elements \( x(k) \) may occur at the right-hand side. A general max-plus linear system thus has the form

\[
x(k) = \bigoplus_{l=0}^{p} (A_l \otimes x(k-l)) \oplus d(k),
\]

where \( p \) is the maximum delay between any two events and \( A_l \) contains the elements with a period delay \( l \) only. Note that this equation equals the first order representation if \( p = 1 \) and \( A_0 \) is the empty matrix (all entries are \( -\infty \)). However, these higher order linear systems can be transformed into an equivalent first-order representation, see e.g. Goverde & Soto y Koelemeyer (2000).

### 2.3 Timed Event Graph

A max-plus linear system can be represented by a timed event graph (or timed marked graph), which is a restricted class of a Petri net. A timed event graph is essentially a
bivalued weighted digraph, where the nodes correspond to events and the arcs to processes between events. The arcs have two weights: a holding time corresponding to the process (or transportation) time and a marking (or token count) associated to the period delay between the event pair. In a first-order model each arc has exactly one token. In a higher order model, a marking ranges between 0 and $p$, where $p$ is the maximum period delay between any two events. The marking can be interpreted as the number of trains running between two stations at a certain reference time.

For the general max-plus linear system (2) the graph is constructed as follows. A node $i$ is defined for each event $x_i$ and an arc exists from a node $j$ to $i$ if the $ij$-th entry of the state matrix $A_l$ ($l = 0, \ldots, p$) is finite, with holding time $(A_l)_{ij}$ and $l$ tokens. The number of arcs is thus the number of entries over all $A_l$ with $(A_l)_{ij} \neq -\infty$. For more detail, see Goverde & Soto y Koelemeijer (2000).

In Petri net terminology the nodes are called transitions and the arcs are replaced by places between the transitions, see e.g. Goverde & Soto y Koelemeijer (2000) and Baccelli et al. (1992).
3.1 Introduction

The max-plus model of the train network consists of two parts: the state matrix constructed from process times between timetable events, train orders, and the allocation of trains to the network, and the timetable vector containing the scheduled event times. Analysis of the state matrix identifies critical circuits and network stability indicators associated to the train network structure (interactions, departure sequences, train allocation). This critical circuit analysis requires computation of the eigenvalue and eigenvector of the matrix $A$ (in max-plus algebra), i.e. solving

$$A \otimes v = \lambda \otimes v.$$  \hspace{1cm} (3)

A very efficient policy iteration algorithm (Cochet-Terrasson et al., 1998) is implemented in PETER to solve eigenvalue problem (3). Even large-scale networks of several 10,000 variables are solved within a few seconds. The eigenvalue $\lambda$ is the minimum network cycle time and the eigenvector $v$ the corresponding timetable. This timetable gives the earliest possible departure times (with respect to the imposed train allocation and train orders) according to which the train network can operate with minimum cycle time. Applying this (feasible) timetable yields the earliest operation mode (EOM) of the train network.

3.2 Minimum Cycle Time

Analysis of the state matrix $A$ gives the minimum cycle time in which the train network could operate. This minimum cycle time equals the average travel time of trains on the critical circuit in the network. This critical circuit is thus the ‘longest’ circuit in the network. The actual cycle time $T$ (usually $T = 60$) should clearly exceed the minimum possible one. This represents a stability test: if $\lambda < T$ then the system is stable. However, if $\lambda$ is close to the cycle time then the system is sensitive to disruptions. Stability here means that there is enough slack on the critical circuit, so that delays will surely settle.

System stability may be improved by changing the sequence of critical processes that make up the critical circuit. Possible actions are changing train orders, adding (or move) a train to one of the critical processes, or decreasing the critical process time by e.g. faster train units, shorter dwell times, shorter transfer times (cross-platform transfers), or infrastructure investments.

In practice, a network timetable contains several components (subnetworks) that are mutually connected only by the periodic timetable. Hence, the minimum cycle time corresponds to the critical circuit(s) of the most critical component. In a well-balanced system the critical cycle times of all components should be comparable, corresponding to an even distribution of trains to the network. PETER therefore computes the critical circuits and cycle times of each component in the train network. The number of components is an indication of the complexity or connectedness of the train system. The smaller this number is, the higher the system complexity. The
associated range of critical cycle times quantify the balance between the various subnetworks.

### 3.3 Network Throughput

Throughput of a periodically operating train network is defined as

$$ \rho = \frac{\lambda}{T}, $$

where $\lambda$ is the minimum cycle time and $T$ the actual cycle time. The throughput $\rho$ is a network performance indicator denoting a trade-off between maximum performance (under ideal circumstances) and robustness. It presents the minimum cycle time relative to the actual cycle time. Obviously, a stable system requires $0 < \rho \leq 1$, where $\rho = 1$ is the saturated case in which the mean cycle time of trains on the critical circuit is just (a multiple of) $T$ minutes. When $\rho < 1$ the system operates below its maximum (theoretical) performance and hence contains buffer times to compensate for delays.

Network throughput refers to the throughput of the critical circuit. A large throughput on this circuit does not necessarily mean that the throughput on any corridor (which can be seen as a sub-network) is also this high. It is therefore also relevant which transportation links make up the critical circuit, and how the critical cycle time relates to the cycle times of train lines, corridors, or other sub-networks. The cycle times, and hence throughputs, of train lines are easily obtained when connections and headway are discarded: the circuits are then exactly the train line circulations (assuming layover times are included).

### 3.4 Stability Margin

The *stability margin* $\Delta$ is a network performance indicator of robustness of the train network. The stability margin is defined as the maximum simultaneous increase of all process times such that the train network can still be operated with cycle time $T$. This margin may differ from the value $T - \lambda$ as another circuit may become critical when the process times are increased.

The stability margin can be computed by solving an auxiliary eigenvalue problem (Goverde & Soto y Koelemeijer, 2001). Let the matrix $B$ be obtained from the state matrices $A_i$ of the system (2) by subtracting $IT$ from each entry and taking the maximum over the matrices, i.e., $\bigoplus_{i=0}^\rho (A_i - I \cdot T)$. Then the stability margin is $\Delta = -\mu$, where $\mu$ is the solution of the eigenvalue problem

$$ B \otimes \nu = \mu \otimes \nu. $$

The stability margin corresponds to the circuit with the least average buffer time. This circuit is not necessarily the same as the critical circuit with the critical cycle time but depends on the amount of running trains in each circuit.
4.1 Introduction

The actual periodic timetable – modelled as the timetable vector – can be obtained from the eigenvector (associated to the earliest operation mode) by additional slack times between events. These slack or buffer times are necessary for robustness of the timetable. A second stage in the network analysis – contemporary to critical circuit analysis – therefore focuses on the distribution of buffer times over the network timetable. This is referred to as recovery time analysis.

4.2 Recovery Times

A timetable generally contains slack to compensate for process time variations. This slack is provided in scheduled process times (e.g. running time margins) or between two directly interacting trains (buffer times). From a network point of view a sequence of train runs embraces an accumulation of slack times. These train sequences include train runs connected by train stops or transfers but also more abstract headway interactions. For any two trains we are interested in the amount of cumulative slack time over a train sequence going from one train to the other. However, in general multiple train sequences can be considered between any pair of trains. Therefore we define the cumulative recovery time from train \( i \) to train \( j \) as the minimum cumulative slack time over all train sequences going from train \( i \) to \( j \) in the train network. Note that this is equivalent to the largest delay of train \( i \) that not reaches train \( j \).

We distinguish between three types of recovery time indicators:

- *delay impact* of a train to all trains over subsequent train sequences,
- *delay sensitivity* of a train w.r.t. delayed trains on incoming train sequences,
- *feedback recovery time* of a train: minimum recovery time over any circuit.

Delay impact refers to possible future delays whereas delay sensitivity depends on past delays that may have propagated over previous train sequences. The feedback recovery time gives the maximum delay of a train that has settled before returning over any train sequence (circuit).

4.3 Recovery Time Matrix

The cumulative recovery times can be represented in a recovery time matrix in which the \( ij \)-th entry is the cumulative recovery time from train \( j \) to train \( i \). The delay impact vector of a train \( i \) equals the \( i \)-th column of the recovery matrix, the delay sensitivity vector of train \( i \) equals the \( i \)-th row of the recovery matrix, and the feedback recovery time vector of all trains is the diagonal of the recovery matrix.

The recovery time matrix can be computed numerically using an all-pair shortest path algorithm on a graph constructed from the state matrix of the max-plus model. This graph is obtained from the original timed marked graph by subtracting from each arc
weight (holding time) the number of periods (tokens) associated to this arc and changing its sign. The resulting arc weights represent the recovery time of the arc. It can be proved that each circuit in this graph has a positive length, by which a shortest path exists between any two strongly-connected nodes.

In PETER, the efficient Johnson repeated single-source shortest path algorithm for sparse networks is implemented (Cormen et al., 1990). This method uses the well-known Dijkstra algorithm as a subroutine, where the priority queue is implemented as a Fibonacci heap.
Also delay propagation can be utilized to analyze timetable sensitivity to sets of initial delays. The max-plus model and an initial delay vector – containing the departure delays at a certain reference time – are used to simulate the delay propagation of initial delays over time and space. In fact, the max-plus (sparse) matrix equation generates a particular efficient single-run discrete-event simulation. The computation time is a fraction of a second, even for large-scale networks.

The initial delay vector summarizes all train delays in the network at a certain reference time. The max-plus model then computes the delay propagation over the network and over future periods. The simulation output is given in different levels of detail. The most refined output gives the delay of each train in terms of the departure delay and its period-of-occurrence. The result can be given in tabular form or in a network visualization where the amount of delay is indicated by a colour scheme.

Aggregated output is more convenient for comparison of different scenarios or distinct timetables. The presented aggregated results are the total secondary delay, number of reached trains, average secondary delay, number of reached stations, and the settling period.

By using nominal process times the emphasis of the delay propagation model is on analyzing the interconnection structure as dictated by the timetable. Computations of many scenarios are achievable to find structural shortcomings in a timetable design or to evaluate the effect of control strategies. Different control strategies result essentially in distinct delay scenarios represented by an initial delay vector. Using the max-plus model the implications of the different delay scenarios are computed and compared. This scenario analysis is useful in deriving all kinds of standard control strategies in case of e.g. maintenance works and track obstructions. Also online decision support systems may benefit from quickly evaluating several delay scenarios to resolve detected conflicts.
6 Case Study: Dutch Intercity Network

The Dutch Intercity (IC) Network of 2000-2001 contains 19 IC train lines serving 70 (IC) stations. This network is obtained from DONs and imported into PETER. The max-plus model consists of 317 departure events (nodes) and 361 line segments (arcs), including 44 connections between train lines. A minimum of 112 trains is necessary to cover all IC train circulations. The model contains 137 tokens, which is a measure of the problem dimension.

The policy iteration algorithm computes 10 critical circuits associated to 10 network components. Figure 1 shows the most critical circuit and its component. The critical circuit contains IC1600 and IC500 trains between Schiphol and Groningen, including 8 stops, 2 transfers at Amersfoort, and 2 turns at Groningen and Schiphol, respectively. The minimum cycle time is 55:12 minutes. The train network is operated according to an hourly timetable, giving a throughput of 92%. The stability margin on this critical circuit is 1:45 minutes.

Figure 1: IC network, its critical circuit (black links) and associated component (white links), and delay sensitivity of IC 1600 Amf-Dvd (dark nodes)
sensitivity of the IC1600 train from Amerstoort to Druvendrecht (the coloured nodes; the darker a node the less recovery time). The preceding stops of the IC1600 train are most critical, where a delay at Apeldoorn has zero recovery time, whereas the robustness is gradually increased over the preceding stations to 4 min recovery time at Enschede. The preceding stops of the feeder IC500 train are next critical, with 4 min recovery time after departure at Zwolle, increasing to 8 min in Groningen and 9 min in Leeuwarden (the trains from Leeuwarden and Groningen are coupled in Zwolle). Moreover, the IC1600 train may get delayed by the IC21600 from Amsterdam CS (11 min), the IC12500 from Rotterdam (13 minutes), and the IC500 from The Hague CS (19 min) and the intermediate stops with decreasing recovery time. This clearly shows the interactions in the most critical network component.

![Diagram of delay propagation over the IC network](image)

**Figure 2: Delay propagation over the IC network**

More detailed insight in the interconnectedness of the IC train services is obtained by studying the delay propagation of initial delays. As an example, we consider a scenario in which during one hour each of the 20 departing IC trains from Utrecht CS has a 10 minute departure delay. Figure 2 shows the resulting delay propagation, where the dark nodes represent delayed departures (the darker the more delay). In total 43 stations are reached, including the terminal stations Groningen, Leeuwarden, and Nijmegen, where the delays propagate over new train circulations unless reserve train units are dispatched. The delays also reach 5 other terminal stations, however at these stations enough buffer time is available to guarantee stable train circulations, i.e., new circulations can be initiated in accordance to the process plan. In the other
directions the delays have been eliminated by the recovery times before reaching the terminal stations. Apart from the 20 initial trains, 13 additional trains get delayed, with an average departure delay of 5:14 min, including trains starting a new circulation. In total 103 departure delays are computed, including the (reduced) delays at successive stops of a train line. Eventually, all delays are eliminated after 3 hours.
Network timetables have an intrinsic structure that can be conveniently modelled using max-plus algebra. Efficient analysis methods can be applied to this max-plus model including critical circuit analysis, shortest path methods, and matrix computations. Using these methods many valuable performance indicators can be computed, including

- network performance indicators (minimum cycle time, throughput, stability margin),
- recovery times (delay impact vectors, delay sensitivity vectors, feedback recovery times),
- delay propagation (total secondary delay, number of delayed trains, average secondary delay, settling time).

The max-plus modelling and analysis techniques have been implemented in the software tool PETER (Performance Evaluation of Timed Events in Railways). The algebraic approach combined with efficient graph theoretic algorithms allows PETER to analyse large-scale networks in real time, which is a major contribution to the traditionally time-consuming railway planning process. PETER has a user-friendly man-machine interface, many edit possibilities, and convenient input formats.

Another feature is the compatibility with DONS – the timetable design tool of the Dutch Railways and Railned. DONS output files can be imported in PETER, which automatically builds the associated max-plus model. All analysis methods in PETER are then available to quickly evaluate timetable structures generated by DONS, which is a major contribution to computer-aided timetable design.

Traditionally, simulation is used to analyse railway operations. However, simulation is a very time-consuming process. Moreover, a difficulty in large-scale network-wide simulation is the large amount of output that can be generated, which disguises the critical outcomes. The analytical performance evaluation tool PETER can be used to quickly reveal the bottlenecks in the timetable design. Detailed stochastic simulation can then be used to zoom into these critical components using specific scenarios or emphasizing a critical sub-network.
Acknowledgement

This publication is a result of the research programme Seamless Multimodal Mobility, carried out within the TRAIL Research School for Transport, Infrastructure and Logistics, and financed by Delft University of Technology.


