VIII Modelling Week UCM (2014)

Efficient Railway Timetabling Using Max-Plus Algebra

Masters in Mathematical Engineering

Participants: Miguel Ambrona Castellanos
Francesco Della Santa
Beatriz Frade Suárez
Jorge González Ortega

Instructor: Ebrahim Patel

Abstract

This paper shows how Max-Plus Algebra (MPA) can be used to improve a railway's timetable. After introducing the problem and presenting the basic results of the MPA in relation to railway systems, the paper analyzes how to use MPA to define an efficient timetable for the central area of the Spanish railway network. Matlab has been used in order to make simulations of the railway network's behaviour.

Contents

1 Introduction: Goals, Rules & Assumptions

Each customer of a railway system wishes an efficient timetable that satisfies his needs. But what does an *efficient railway timetable* mean?

Obviously, to be efficient, a timetable should offer travellers a frequency of departures as high as possible, but this is not enough. In fact, a good timetable, organizes its departures cleverly to be regular, so that customers can remember them easily. Finally, an efficient railway system, should have a timetable that is able to manage delays, in order to reduce customers' discomforts.

Using this analysis on what an efficient timetable ought to provide, it is possible to summarize the goals our solution should achieve:

G1 Frequency of the trains as high as possible.

G2 Regular departure times.

G3 Flexibility to manage delays.

Appart from these goals, two main rules are considered necessary:

R1 Trains should wait for each other to allow the changeover of passengers.

R2 All trains at a station depart at the same time as soon as they are allowed.

After determining our goals and rules, it is possible to set some assumptions through the observation and simplification of a railway system. First of all, it is easy to notice that, in every railway system, a customer can travel from any station to another one. Therefore, since a graph is used to describe a railway system[1], the graph will have to be *strongly connected*. Finally, the considered model is simplified so that there is only one train per track and travel times are fixed. The assumptions are:

A1 The railway system's associated graph is strongly connected.

A2 There is one train per arc of the graph.[2]

A3 Travel times are fixed.

---

[1] Where nodes are stations, arcs are railways and arcs' weights are travel times.

[2] An *adding-one-train-to-a-track* operation will later be explained so that the new network's graph still follows this rule.

2 Well Known Results of MPA & Applications to a Railway System

In this section, some definitions and basic results of Max-Plus Algebra's (MPA's) theory are introduced.

2.1 Notation & Operators

In MPA there are two main operators (which give its name to this kind of Algebra): the *max* and *plus* operators. These operators are respectively denoted with the *o-plus* ($\oplus$) and the *o-times* ($\otimes$) symbols and used as in the following example:

Example 2.1 *Let* $a_i \in \mathbb{R},\ \forall\, i \in \{1, \dots, n\} \subset \mathbb{N}$*, then:*

$$a_1 \oplus a_2 \oplus \cdots \oplus a_n = \bigoplus (a_1, \dots, a_n) = \bigoplus_{i=1}^{n} a_i = \max (a_1, \dots, a_n)$$

$$a_1 \otimes a_2 \otimes \cdots \otimes a_n = \bigotimes (a_1, \dots, a_n) = \bigotimes_{i=1}^{n} a_i = \sum_{i=1}^{n} a_i$$

Observation 2.1 *It should be remarked that the* neutral element *of the* o-plus *operation is* $\varepsilon := -\infty$*. In the case of the* o-times *operation,* ε *is the* absorbing element *and* 0 *the* neutral *one.*

The *o-plus* and *o-times* operators may be regarded respectively as the *sum* and *product* of linear Algebra. Therefore, *max* and *plus* operations may be extended to matrix operations following rules similar to those of the sum and product of matrices. In order to better understand these matrix operations in MPA we present the next examples, which also introduce the *o-power* notation for matrices:

Example 2.2

$$\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \otimes \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} (1 \otimes 2) \oplus (2 \otimes 3) \\ (2 \otimes 2) \oplus (3 \otimes 3) \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \otimes \begin{pmatrix} 0 & \varepsilon \\ \varepsilon & 0 \end{pmatrix} = \begin{pmatrix} (1 \otimes 0) \oplus (2 \otimes \varepsilon) & (1 \otimes \varepsilon) \oplus (2 \otimes 0) \\ (2 \otimes 0) \oplus (3 \otimes \varepsilon) & (2 \otimes \varepsilon) \oplus (3 \otimes 0) \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}^{\otimes 2} := \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \otimes \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} = \begin{pmatrix} (1 \otimes 1) \oplus (2 \otimes 2) & (1 \otimes 2) \oplus (2 \otimes 3) \\ (2 \otimes 1) \oplus (3 \otimes 2) & (2 \otimes 2) \oplus (3 \otimes 3) \end{pmatrix} = \begin{pmatrix} 4 & 5 \\ 5 & 6 \end{pmatrix}$$

The second of the previous examples includes the identity matrix $I_2 \in \mathbb{R}^{2\times 2}$ for the *o-product*. This matrix is easily generalized to n dimensions.

Once the new operators and notions have been defined, we begin to realize the advantages of MPA. This way of understanding the task converts the non-linear problem of finding an efficient timetable into some sort of linear problem, sharing many well-known concepts with conventional linear Algebra.

2.2 Railways & Graph Theory

As mentioned in Section 1, railways are represented by graphs with a node per station and an (oriented) arc for each track between two stations with the arcs' weights representing the travel times. Let's start with an example to make this clearer.

Example 2.3 *In Figure 1, the graph of a simple railway system has been represented. There are only two stations and four tracks where the time-travel is described by the arcs' weights. Due to assumption A2, there are only four trains, one per track.*

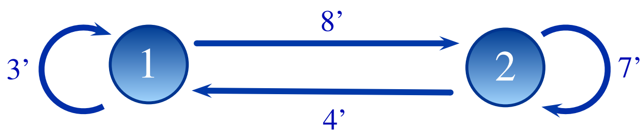

Figure 1: Example of a simple railway system's graph

Let $x(0) \in \mathbb{R}^2$ be the initial departure time of the trains at the stations, where $x_i(0)$ is the departure time of S_i's trains $\forall i \in \{1,2\}$ (remember that all trains at the same station have the same departure time due to rule R2).

According to rule R1, trains departing from one station must wait for the rest in order to allow the changeover of passengers. Thus, next departure time $x(1)$ will have components $x_i(1)$ such that:

$$x(1) = \begin{pmatrix} x_1(1) \\ x_2(1) \end{pmatrix} \geq \begin{pmatrix} \max\{x_1(0)+3, x_2(0)+4\} \\ \max\{x_1(0)+8, x_2(0)+7\} \end{pmatrix} = \begin{pmatrix} (x_1(0)\otimes 3) \oplus (x_2(0)\otimes 4) \\ (x_1(0)\otimes 8) \oplus (x_2(0)\otimes 7) \end{pmatrix}$$

As trains have to leave the station as soon as possible (goal G1), the inequality must indeed be an equality and we may write:

$$x(1) = \begin{pmatrix} (x_1(0)\otimes 3) \oplus (x_2(0)\otimes 4) \\ (x_1(0)\otimes 8) \oplus (x_2(0)\otimes 7) \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 8 & 7 \end{pmatrix} \otimes x(0) =: A \otimes x(0)$$

Where A is the adjacency matrix *of the graph in Figure 1 whose elements $a_{i,j}$ represent the travel time from S_j to S_i.*

In a similar way, the k-th departure time ($k > 0$) may be described by the next equation:

$$x(k) = A \otimes x(k-1)$$

If we suppose we have $x(0) = (0,0)^t$, then it is possible to write the associated efficient timetable:

$x(0)$	$x(1)$	$x(2)$	$x(3)$	$\cdots$
0	4	12	19	$\cdots$
0	8	15	22	$\cdots$

The most relevant results developed in the above example are indicated now:

Proposition 2.1 *Let G be a strongly connected graph representing a railway system, $A \in \mathbb{R}^{n\times n}$ its adjacency matrix*[3] *and $x(0) \in \mathbb{R}^n$ the vector of initial departure times. Then, the k-th departure times are given by:*

$$x(k) = A \otimes x(k-1) \tag{1}$$

Corollary 2.1 *Another way to describe the k-th departure time vector is:*

$$x(k) = A^{\otimes k} \otimes x(0) \tag{2}$$

2.3 Eigenvalues & Eigenvectors

Given a square matrix A, as in conventional linear Algebra, we can think of eigenvalues and eigenvectors, λ and x, such that:

$$A \otimes x = \lambda \otimes x$$

[3] As G is strongly connected, we say A is *irreducible*.

However, in this case, the notation $\lambda \otimes x$ refers to a vector with the same length as x whose i-th element equals $\lambda \otimes x_i = \lambda + x_i$. If such a λ exists, it is called an *eigenvalue* of matrix A and x is a corresponding *eigenvector*.

Now, let $x(0)$ be an eigenvector of A corresponding to the eigenvalue λ, then we have:

$$x(1) = A \otimes x(0) = \lambda \otimes x(0)$$
$$x(2) = A \otimes x(1) = A \otimes (\lambda \otimes x(0)) = \lambda^{\otimes 2} \otimes x(0)$$

In general, we can write the k-th departure time like this:

$$x(k) = \lambda^{\otimes k} \otimes x(0) \tag{3}$$

Note that $\lambda^{\otimes k}$ in MPA is equal to $k \times \lambda$ in conventional Algebra. Additionally, eigenvectors are not unique. If we add a constant to all elements in an eigenvector, the resulting vector will be an eigenvector too.

Example 2.4 *Considering the matrix associated to the graph in Figure 1, we have:*

$$\begin{pmatrix} 3 & 4 \\ 8 & 7 \end{pmatrix} \otimes \begin{pmatrix} \alpha \\ 3+\alpha \end{pmatrix} = 7 \otimes \begin{pmatrix} \alpha \\ 3+\alpha \end{pmatrix}$$

Where $\alpha \in \mathbb{R}$ has an arbitrary value. Therefore, the matrix has eigenvalue 7 and an infinite number of eigenvalues, one for each value of α.

In the railway context, eigenvalues and eigenvectors are important concepts because if we set the initial condition of the system in (2) to an eigenvector, the evolution of the sequence will be regular.

Example 2.5 *Assuming we have $x(0) = (0,3)^t$ (an eigenvector for the matrix in Figure 1). Thus, the associated timetable becomes regular and trains depart every seven time units from both stations:*

$x(0)$	$x(1)$	$x(2)$	$x(3)$	$\cdots$
0	7	14	21	$\cdots$
3	10	17	24	$\cdots$

2.4 Cyclicity & Eigenspace

In this section, we define an important concept: the *cyclicity*.

Definition 2.1 *Let G be a graph and let A be the associated matrix. We define the* cyclicity *of G, σ_G, in the following way:*

- *If G is strongly connected, σ_G equals the greatest common divisor of the lengths of all elementary circuits*[4] *in G.*
- *If G is not strongly connected, σ_G is the least common multiple of the cyclicities of all maximal strongly connected subgraphs*[5] *of G.*

Example 2.6 *Let's consider the graph G in Figure 2:*

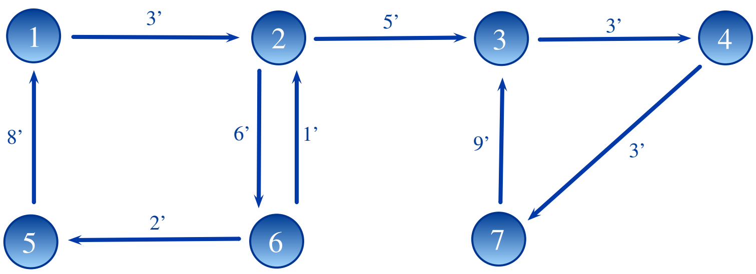

Figure 2: Example of cyclicity in a graph

It is easy to see that G is not strongly connected. For example, it is impossible to travel from node 7 to node 2. However, G contains strongly connected subgraphs.

We can consider the strongly connected subgraph composed of nodes 1, 2, 5 and 6 and all arcs between those nodes. As this graph is not contained in any other strongly connected subgraph, it is maximal (note that there is at least one strongly connected subgraph contained in it, the circuit $\{2, 6, 2\}$).

The considered subgraph has two elementary circuits in it: $\{1, 2, 6, 5, 1\}$ of length 4 and $\{2, 6, 2\}$ of length 2. If we apply Definition 2.1, the cyclicity of the subgraph is $\gcd(4, 2) = 2$.

[4] A circuit is said to be elementary if there is no repetition of nodes.

[5] A strongly connected subgraph is said to be *maximal* if it is not contained in another strongly connected graph.

There is yet another maximal strongly connected subgraph in G, formed by the nodes 3, 4 and 7 and all arcs between them, whose cyclicity is 3. Knowing this, we can compute the ciclycity of graph G, which is $\mathrm{lcm}(2,3)$, *i.e.,* 6.

We introduce now an important theorem, that says that if A is irreducible, the eigenvalue is unique:

Theorem 2.2 *Let G be a graph and let A be its associated matrix. If A is irreducible, or equivalently if G is strongly connected, there is one and just one eigenvalue (but possibly several eigenvectors). This eigenvalue is equal to the maximal average weight of all circuits in the graph:*

$$\lambda = \max_{c \in \mathcal{C}} \frac{|c|_\omega}{|c|_l} \tag{4}$$

Where $\mathcal{C}$ is the set of all circuits of G, $|c|_\omega$ is the total weight of the circuit c and $|c|_l$ is the number of arcs in the circuit c.

Definition 2.2 *A circuit c is said to be* critical *if its average weight is maximal, i.e.,* $\lambda = \frac{|c|_\omega}{|c|_l}$.

Definition 2.3 *The subgraph of all nodes and arcs belonging to a critical circuit is called* critical graph.

In Figure 2, it is easy to determine all circuits. There are three elementary circuits: $\{1,2,6,5,1\}$, $\{2,6,2\}$, $\{3,4,7,3\}$. Being their average weights 4.75, 3.5 and 5 respectively, the critical circuit is $\{3,4,7,3\}$.

We need a method for calculating automatically the eigenvalue of an irreducible matrix A . In general, it is not easy to identify all elementary circuits in a graph and that is why we present the following definitions.

Definition 2.4 *Let A be a square matrix. We consider, for $k \geq 0$, the set of vectors $x(k), x(k+1), x(k+2)\ldots$ where $x(m) = A^{\otimes m} x(0)$, $\forall m \geq 0$. This set is called a* periodic regime *if there exist a value μ and a finite number ρ such that:*

$$x(k+\rho) = \mu \otimes x(k) \tag{5}$$

In that case, the period *of the regime is ρ and $\frac{\mu}{\rho}$ is the* cycletime.

In a railway timetable, the cycletime can be interpreted as the average inter-departure time. In fact, it is not necessary to initialise the system with an eigenvector to ensure periodicity.

Definition 2.5 *Let A be the irreducible matrix associated to a strongly connected graph G. We define the* cyclicity *of A, σ_A, as the cyclicity of its critical graph.*

Theorem 2.3 *Let A be an irreducible matrix with eigenvalue λ and cyclicity σ_A. Then there exists a t such that:*

$$A^{\otimes(k+\sigma_A)} = \lambda^{\otimes\sigma_A} \otimes A^{\otimes k} \quad \forall k \geq t \tag{6}$$

The value t is called the transient time to periodic behaviour.

Note that if we consider the conventional linear Algebra difference between matrices $A^{\otimes(k+\sigma_A)}$ and $A^{\otimes k}$ for $k \geq t$ (being t the transient time), we obtain the next constant matrix:

$$A^{\otimes(k+\sigma_A)} - A^{\otimes k} = \begin{pmatrix} \lambda\,\sigma_A & \cdots & \lambda\,\sigma_A \\ \vdots & \ddots & \vdots \\ \lambda\,\sigma_A & \cdots & \lambda\,\sigma_A \end{pmatrix}$$

Using all the previous results, we have designed the following algorithm:

Require: A (an irreducible square matrix)
Ensure: λ (eigenvalue of A), σ_A (cyclicity) and t (transient time)

```
for pow = 2 to max_pow do
    for dif = 1 to min(max_dif, pow − 1) do

        M ← A^{⊗pow} − A^{⊗(pow−dif)}
        if M is a constant matrix then
            λ ← m11/dif (all elements in M are equal to m11)
            σA ← dif
            t ← pow − dif
            return λ, σA and t
        end if

    end for
end for
```

Algorithm 1: Algorithm for calculating eigenvalues.

With an efficient implementation of this algorithm, we will be able to calculate the eigenvalue λ of most real railway networks. It we do not obtain it, we just have to increment the max_pow and max_dif parameters.

We continue with a useful proposition for computing eigenvectors:

Proposition 2.2 *Let G be a graph with associated matrix A and $\lambda < \infty$ the maximal average circuit weight in A (thus, an eigenvalue of A).*

Let's consider the matrix A_λ with elements:

$$[A_\lambda]_{i,j} = a_{i,j} - \lambda \tag{7}$$

We say A_λ is the normalized *matrix of A and we define:*

$$A_\lambda^* = \bigoplus_{k \geq 0} A_\lambda^{\otimes k} \tag{8}$$

Then, for any node i in the critical graph of G, the i-th column of A_λ^* is an eigenvector of A associated to λ.

Note that, although (8) is a max-plus sum to infinity, in practice it is usually possible to obtain A_λ^* by just computing a few powers of A_λ. The following algorithm uses this idea for computing eigenvectors:

Require: A (an irreducible square matrix)
Ensure: A_λ^* (eigenvectors matrix)

$\lambda \leftarrow$ eigenvalue of A
$A_\lambda \leftarrow A - \lambda$
$A_\lambda^* \leftarrow$ identity matrix (elements $-\infty$ and diagonal 0)

for $k = 1$ **to** *bound* **do**

$A_\lambda^* \leftarrow max(A_\lambda^*, A_\lambda^{\otimes k})$

end for

Algorithm 2: Algorithm for calculating eigenvectors.

It is necessary to fix a *bound* for stopping the loop. We propose the value $t + \sigma_A + 1$, which we have tried with good results, but we do not know if that number is big enough in general as no formal proof has been made.

3 The Spanish Railway Network

We wanted to use all the previous theory in a real case. Therefore, we decided to delimit the central area of the Spanish railway network to work on it. Figure 4 from the Renfe's webpage shows the part of the network we chose:

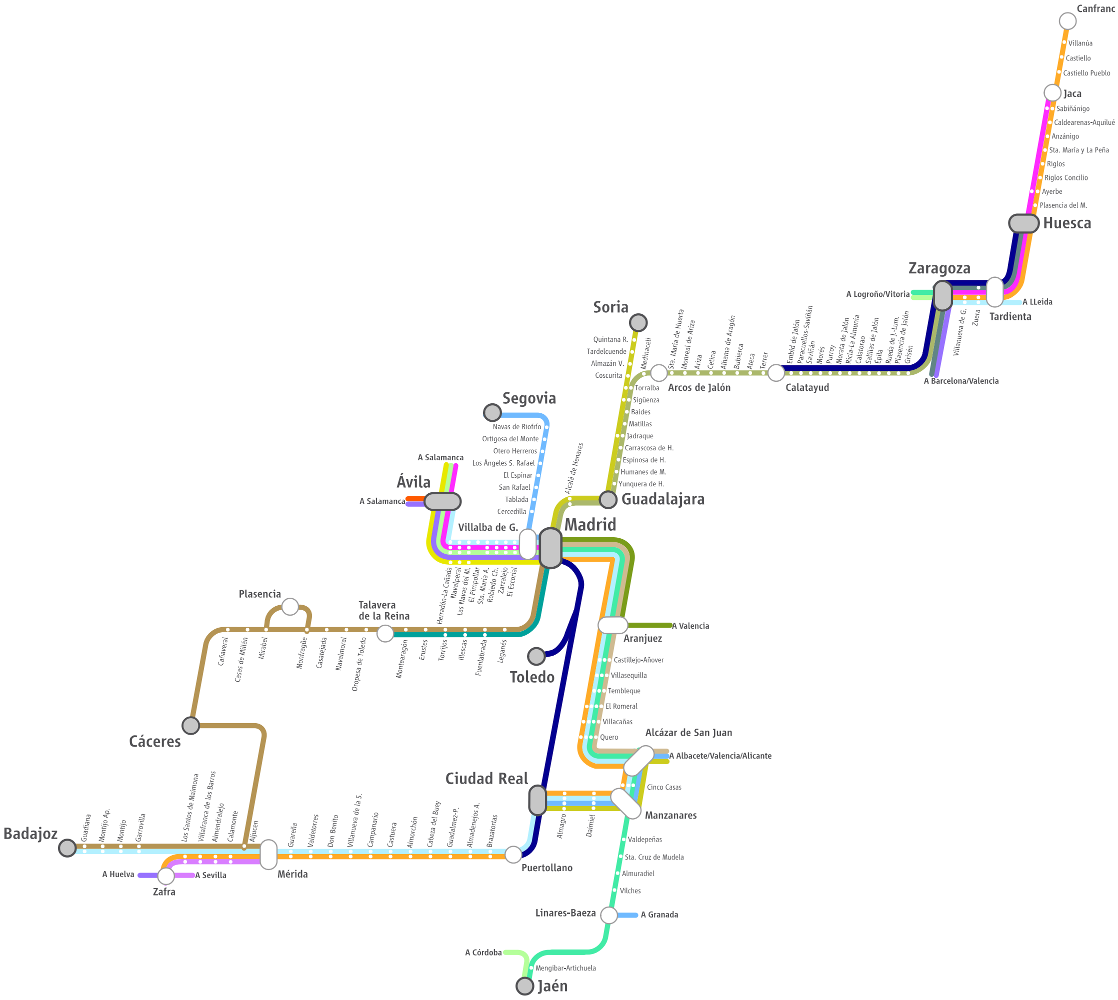

Figure 3: Spanish Railway Map

3.1 Building a Mathematical Model

First of all, we need to turn the map into a graph. This step is really important, because most of the stations in the map are not relevant for our target. We just need those in the end of a line or the ones that allow a changeover.

In order to drop irrelevant stations out of our model, we must delete stations such as *Cáceres* or *Arcos de Jalón* because they are mid-line stations, but we must keep stations such as *Madrid* (in a changeover) or *Badajoz* (in the end of a line). That way, we obtain the following associated mathematical graph:

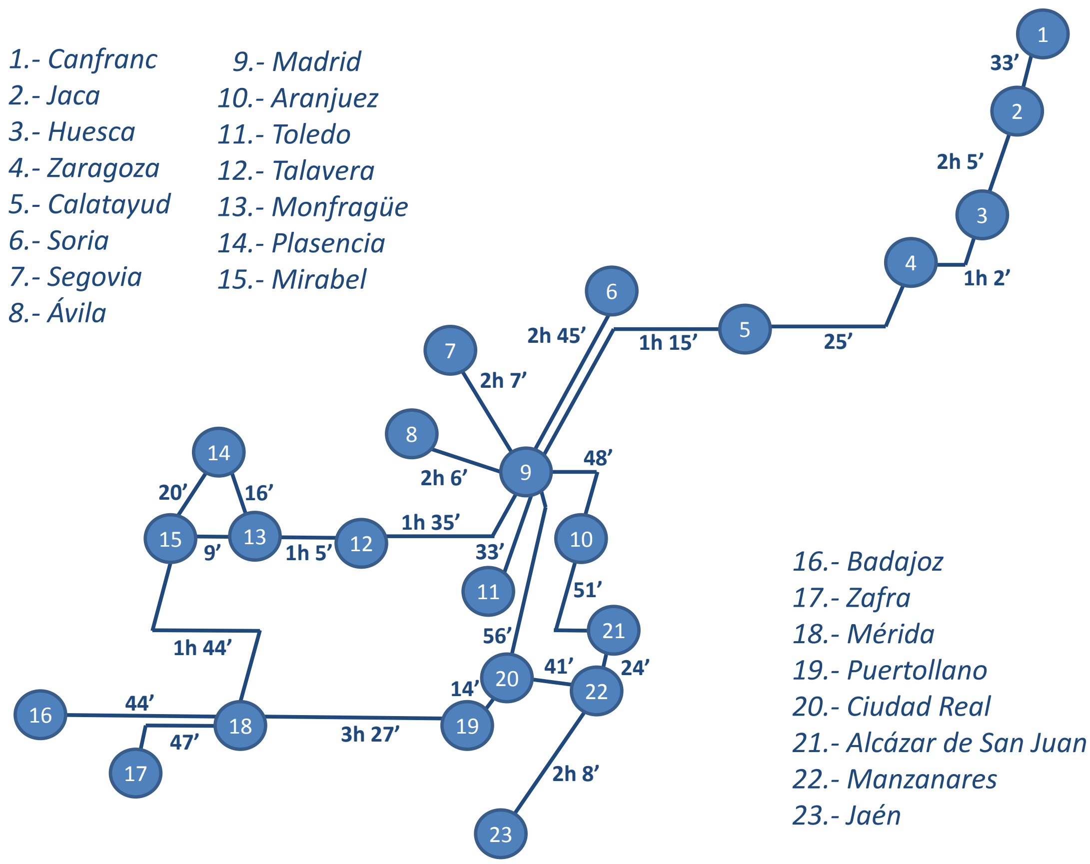

Figure 4: Spanish Network Graph

It must be clarified that every edge in the figure represents two arcs (departure and arrival). Arcs' weights have been estimated thanks to Renfe's travelling application.

3.2 Improving the Network

We have implemented in Matlab the algorithms in Section 2 and calculated the eigenvalue of the Spanish network's associated matrix. Our λ is equal to 207 minutes and it is easy to see that its critical circuit corresponds to the double-way track between stations 18 and 19 (3 h 27' are 207 minutes). This means that the network's lowest inter-departure time is of 3 hours and 27 minutes, which is not at all satisfactory.

We want to improve our network and we propose two different ways of doing it:

- **Adding new trains:**

  Note that the number of arcs in the network is 50. That means that there will initially be 50 trains, one per arc. Clearly, adding trains to a track is a way of improving the frequency of the network, but how can a new train in the Max-Plus Algebra context be added? We will show it in the next example:

  Example 3.1 *Let's consider the network with two trains:*

  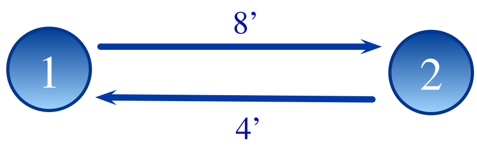

  Figure 5: Example graph of two trains

  From Theorem 2.2 and having only one circuit, we deduce that $\lambda = \frac{8+4}{2} = 6$.

  We will "add a new train" by creating a dummy station*:*

  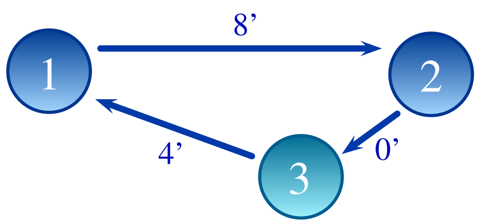

  Figure 6: Example of a dummy station

  As the figure shows, we introduced a new node in the arc connecting nodes 2 with 1 and distributed (arbitrarily) the original arc's weight between the two new arcs.

  We can consider now that there are three trains travelling between stations 1 and 2 (using the three existing arcs). Thus, we have a new eigenvalue, namely $\lambda = \frac{8+4+0}{3} = 4$. *Of course, just as expected, it is lower than the first one.*

- **Mirroring stations:**

  Rule R1 (see Section 1) is that "trains should wait for each other to allow the changeover of passengers". Intuitively, this idea improves our train frequency. However, we can relax this rule in some stations, for instance the ones with a lot of incoming trains, as it doesn't always make sense to wait for all of them. Next example shows how to do this:

  Example 3.2 *Let's consider the graph in Figure 7:*

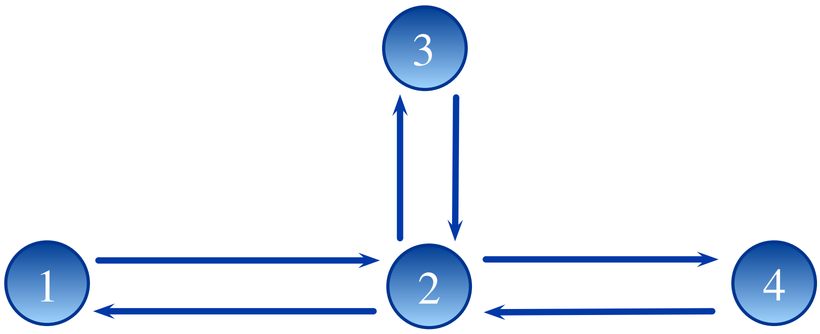

Figure 7: Example for mirroring stations

In station 2, trains must wait for trains coming from stations 1, 3 and 4 before departing. We can relax this condition by creating a mirror station *of station 2:*

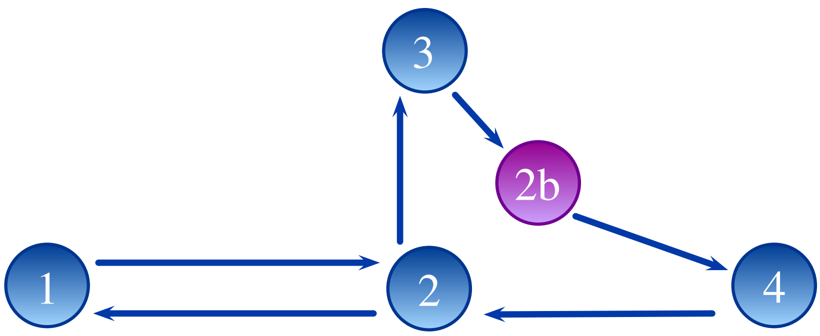

Figure 8: Example of a mirror station

In Figure 8, trains travelling from station 2 to 3 must just wait for trains coming from 1 and 4 (not from 3) and trains going from station 2 to 4 must only wait for trains coming from 3.

Clearly, this technique will increase the frequency of trains as less restrictions are taken into account in the timetable's design.

3.3 Changes in the Spanish Railway Network

It is quite difficult to design an algorithm for adding mirror stations automatically, so we decided manually where to add them. After some considerations on which stations had more incoming trains or just incoming trains that took quite a lot of time to cover their track, we decided which stations to mirror and our network became the following one:

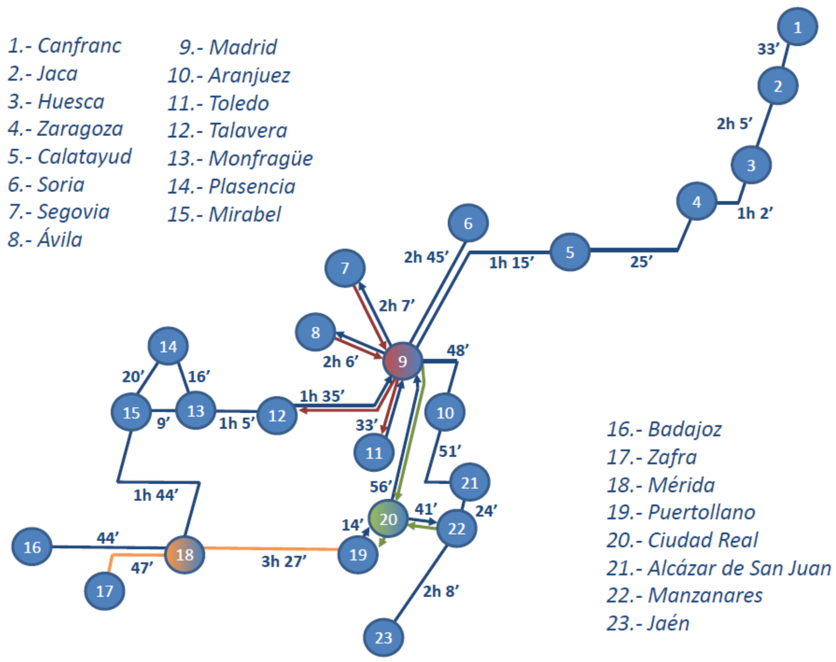

Figure 9: Spanish Railway Map

Three important stations, namely *Madrid*, *Mérida* and *Ciudad Real*, were mirrored.

On the other hand, we implemented in Matlab a recursive function for adding new trains automatically. The function just looks for the critical circuit in the graph and adds a new train in it, then continues by looking for the critical circuit in the new graph and so on. The process ends when λ is lower than a chosen bound.

In both networks, with and without mirror stations, we executed our adding-stations function until λ was under 55 minutes. Figure 10 shows how many trains are needed for reaching the sequence of λ's:

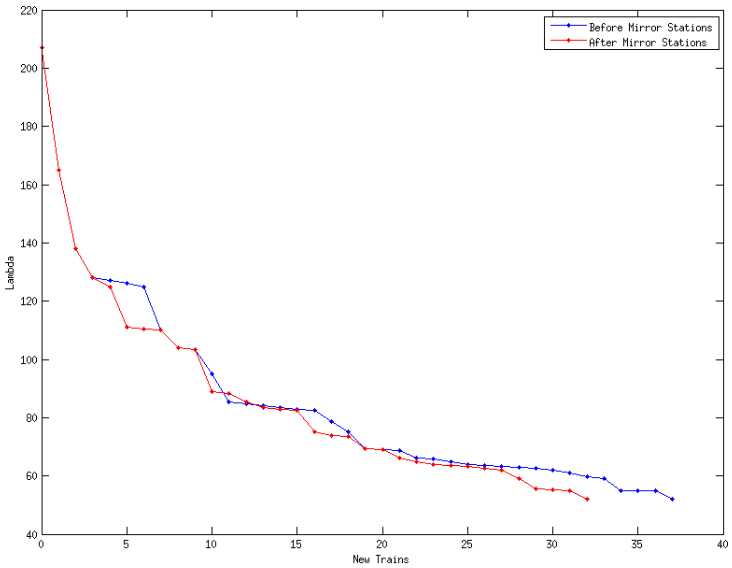

Figure 10: Spanish Railway Map

As we anticipated, the network with mirror stations seems to be better. The red line (network with mirror stations) is most of the time under the blue one (network before adding the mirror stations), which means that we need fewer trains[6] for reaching the same λ.

3.4 Managing Delays

We improved our Spanish railway network by mirroring stations and adding trains. Three mirror stations were considered (*Madrid*, *Mérida* and *Ciudad Real*) and 32 new trains were introduced for reducing the eigenvalue λ from 207 to 55 minutes. In this section, we study what happens with delays if we consider different inter-departure times T to adjust the frequency of trains. Note that $T \geq \lambda$ as λ is the minimal inter-departure time.

In order to show the impact of delays in the network, a Matlab programming code was written for simulating the behaviour of trains in a regular railway system with fixed initial conditions. The graph is represented in **blue** , whilst trains are colored in **orange** when on time and in **red** if delayed.

[6] Trains are expensive, so we want the number of trains to be as small as possible.

First we will consider a tight model, that is, with no difference between the highest frequency we can have and the one we do have (between λ and T). Therefore, let's consider $T = \lambda = 55$ minutes. In Figure 11 a delay of 2 hours in *Zaragoza* is induced:

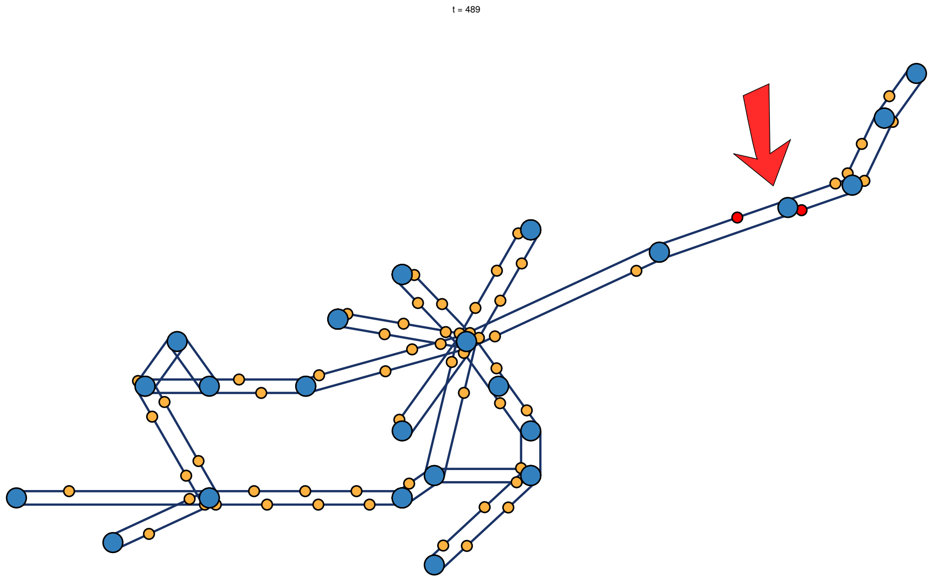

Figure 11: Delay of 2 hours in *Zaragoza*

Figure 12 shows how the delay has propagated after 3 and a half hours:

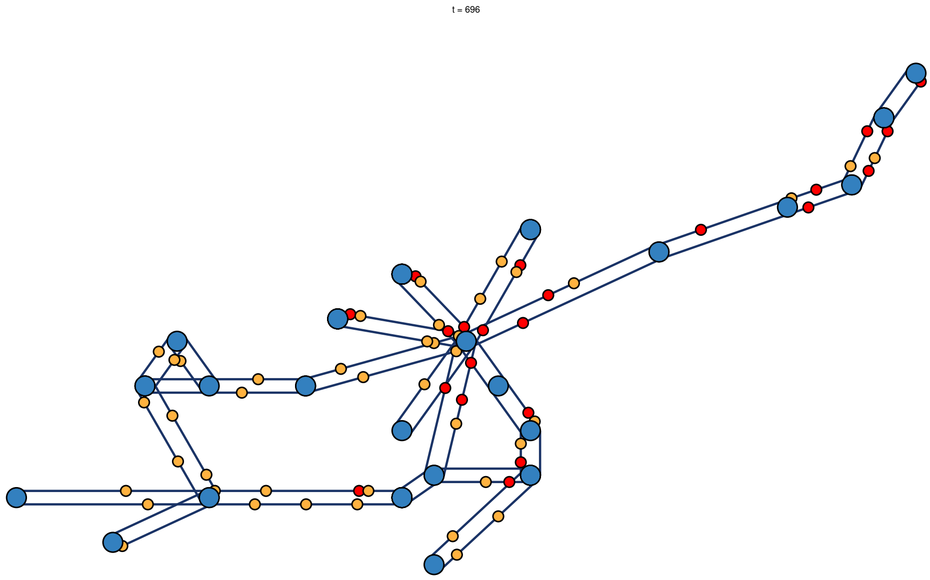

Figure 12: Propagation of the delay in *Zaragoza*

We have a tight model, so the delay will not fade away. This has dramatic consequences because the whole network will be delayed in less than 12 hours:

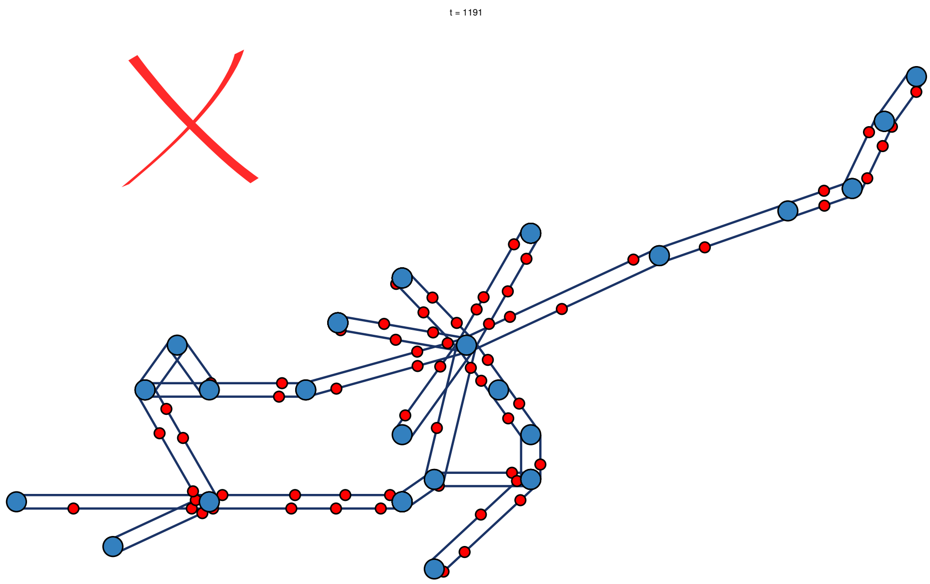

Figure 13: Whole network is delayed

A delay in a tight model will never settle in finite time, even though it may be just located in a particular region of the network. Thus, if we want to manage delays, we need a *stable* timetable, which is achieved by stablishing a value of T strictly greater than λ.

In our final model we choose $T = 60$ minutes. Not just because it is greater than $\lambda = 55$ minutes, but also because it is easy for costumers to remember.

This inter-departure time will allow us to manage delays as we can see in the upcoming images by inducing the same delay of 2 hours in Zaragoza as before:

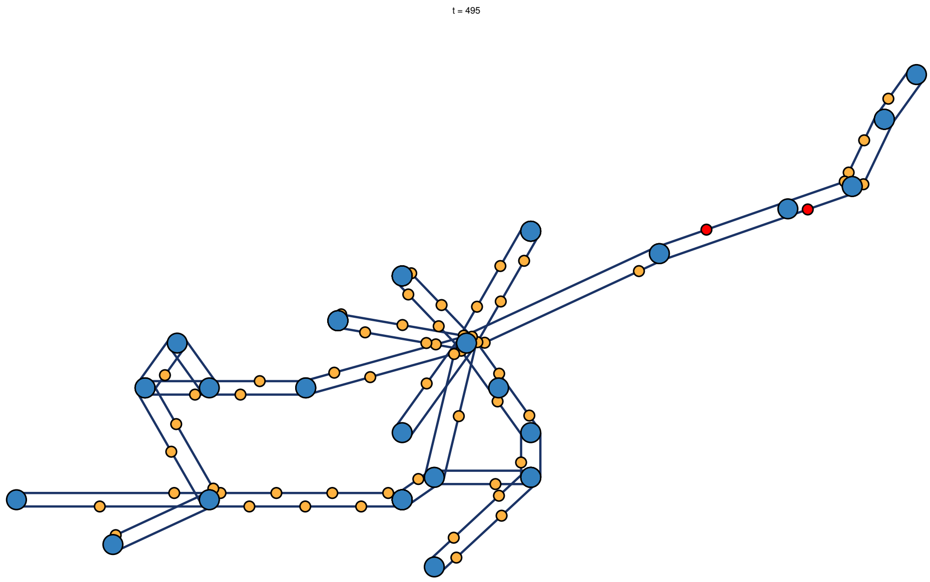

Figure 14: Delay of 2 hours in *Zaragoza*

The propagation of the delay in the network after 2 hours is showed next:

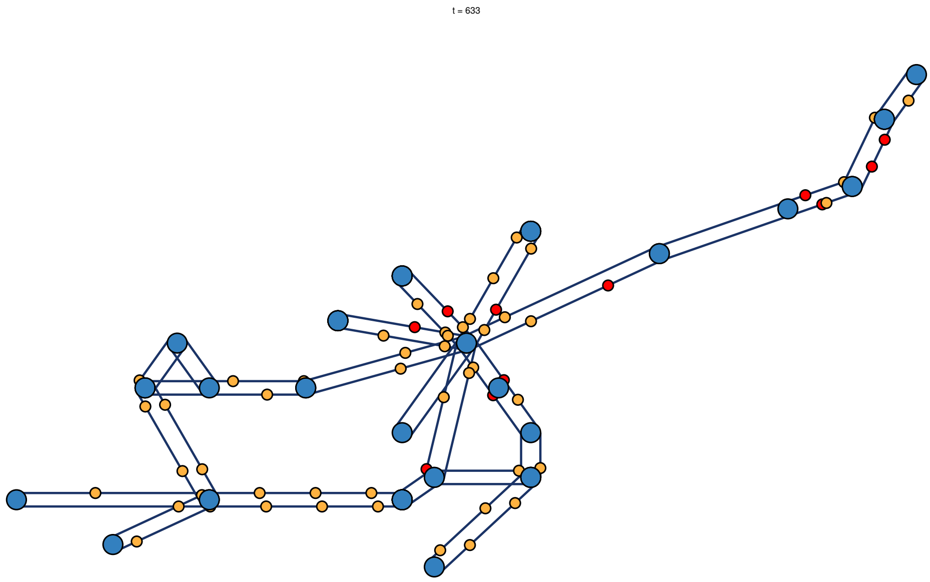

Figure 15: Propagation of the delay in *Zaragoza*

Thanks to the gap between T and λ, the delay will start to disappear after 5 hours, as we can see in Figure 16:

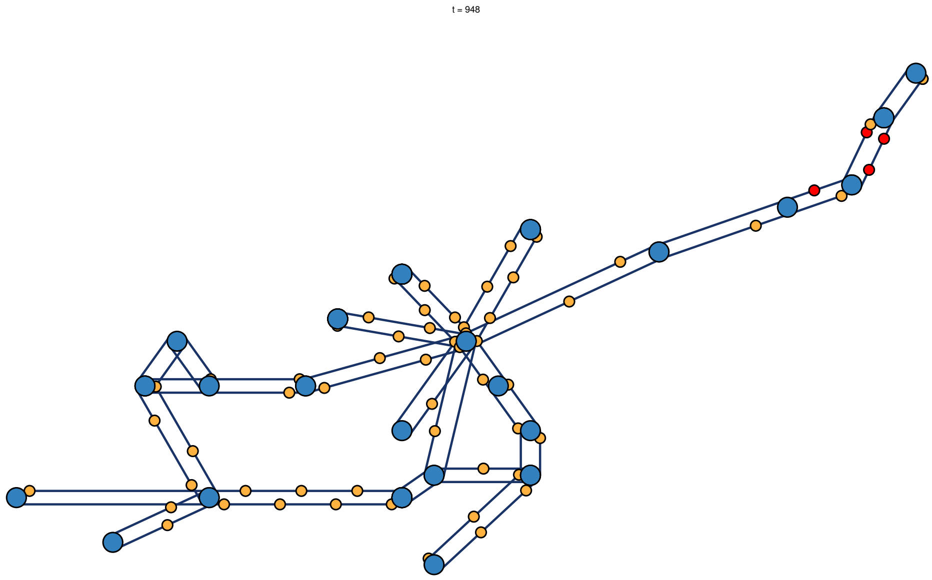

Figure 16: The delay is disappearing

Finally, in Figure 17 the delay is out of the network, just about the time when in the previous example the whole network was completely delayed:

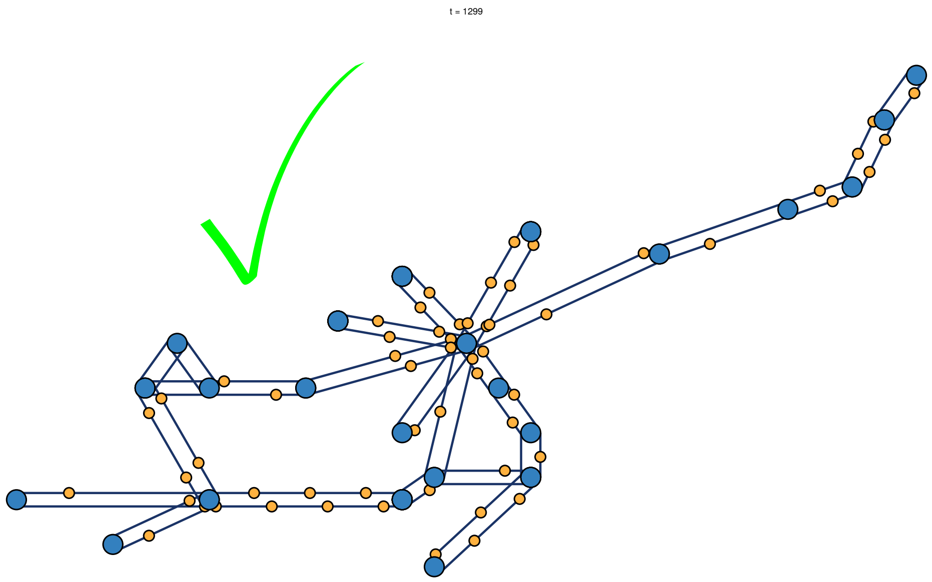

Figure 17: The delay is out of the network

4 Conclusions

Turning our look back to the goals we decided our efficient timetables should achieve, we can be proud as all of them have been handled in some way:

G1 Frequency of the trains as high as possible (thanks to mirror stations and adding new trains).

G2 Regular departure times (thanks to MPA as the initial departure time is an eigenvector).

G3 Flexibility to manage delays (thanks to choosing a larger period than the eigenvalue).

However, in only one week, not all the ramifications of the problem were assessed. A sort list of further work suggestions that could be considered is:

- Think about the possibility of reducing interdeparture times during the rush hour.
- Include the case of trains that have to travel through more than one main station.
- Consider all the regions of Spain together.
- Analize the cost of improvements such as faster trains, new railway tracks...
- Try to apply the delayed model to the study of epidemic models.

Hopefully, more time will be dedicated in the future to study (at least) some of the included ideas.

References

[1] Heidergott, B., Olsder, G.J. and van der Woude, J.W. (2006), *"Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and its Applications"*. Princeton Series in Applied Mathematics.

[2] Patel, E.L. (2014), *"Max Plus at Work: Introduction, Spectral theory and periodic behaviour, delays and stability analysis"*. Working papers.

[3] Web page: http://www.renfe.com/.