Timetable compression using max-plus automata applied to large railway networks

Uyttendaele, Jolien; Van Hoeck, Inneke; Besinovic, Nikola; Vansteenwegen, Pieter

DOI
10.1007/s11750-022-00641-5

Publication date
2022

Document Version
Final published version

Published in
TOP

TOP
https://doi.org/10.1007/s11750-022-00641-5

ORIGINAL PAPER

Timetable compression using max-plus automata applied to large railway networks

Jolien Uyttendaele[1] · Inneke Van Hoeck[1] · Nikola Besinovic[2] · Pieter Vansteenwegen[1]

Received: 3 February 2022 / Accepted: 10 August 2022

Abstract

Demand for railway transportation keeps on growing. Therefore, a thorough understanding of the capacity of railway networks is crucial. In this paper, the well-known compression method based on max-plus algebra is extended. A number of challenges are addressed to apply this compression method to large and complex networks, such as the one considered in this paper. Some trains have to be split artificially, while keeping the parts together during the compression. The trains should also be ordered explicitly, since there is no part of the infrastructure used by all trains. The results in this paper indicate that it is possible to thoroughly analyse the capacity by the adjusted compression method for large and complex networks, but the results should be interpreted with care. The results show, for instance, that the capacity occupation heavily depends on the size of the network that is considered and that it is not easy to give a clear, practical interpretation of the capacity occupation. Nevertheless, the method allowed to determine a number of critical paths and, even more importantly, a number of critical resources in the zones considered.

Keywords Railway capacity analysis · Max-plus algebra · Capacity occupation · Rail capacity

Mathematics Subject Classification 90B06: Transportation · Logistics

✉ Inneke Van Hoeck
Inneke.Vanhoeck@kuleuven.be

[1] KU Leuven Institute for Mobility-CIB, KU Leuven, Celestijnenlaan 300, Box 2422, 3001 Leuven, Belgium

[2] Department of Transport and Planning, TU Delft, Stevinweg 1, 2628 CN Delft, The Netherlands

Published online: 10 September 2022

1 Introduction

The demand for public transport keeps growing due to several economic and environmental reasons. Railway transport will keep on playing a major role in medium- to long-distance transportation. To answer this growing demand, a thorough understanding of the current capacity of railway networks is necessary. This can lead to more efficient planning by identifying bottlenecks. Abril et al. (2008) describe the goal of capacity analysis as "determining the maximum number of trains that would be able to operate on a given railway infrastructure, during a specific time interval, given the operational conditions".

One measure to quantify capacity is capacity occupation. This is the shortest possible time that a sequence of trains needs to drive through a network. Timetable compression is a well-known method to determine capacity occupation (UIC 2013). However, on the microscopic level, this is typically only applied to smaller parts of a network, such as a line or a node. Unfortunately, this does not consider the dependencies in the network and therefore does not give a correct representation of the capacity occupation. Therefore, we will extend the max-plus automata model for timetable compression (Besinovic et al. 2017) to large and complex networks on a microscopic level.

The main contributions of this paper are as follows. First, the max-plus automata method is extended to be applicable to large networks with microscopic detail. The most important extension is a method for determining an appropriate train order that holds for the entire network by splitting trains when required. To obtain a correct compression based on such a train order, dummy resources may be necessary. A second contribution is an adjustment to the existing max-plus algorithm to significantly speed up computation times and easily identify the critical resources in the network. Finally, interesting observations are made regarding the interpretation of the capacity occupation by performing experiments with a varying number of trains.

The paper is structured as follows. In the next section, we discuss the state of the art related to railway capacity analysis using timetable compression. In Sect. 3, we discuss the definitions used in the paper. In Sect. 4, the methodology used to analyse the railway capacity in this paper is explained. This includes a detailed explanation of the max-plus automata, with a focus on the specific extensions that are necessary when analysing a large network. Section 5 introduces the different network instances that we will use for the experiments and an extensive case study. The results of these experiments and the case study are presented in Sect. 6 and Sect. 7 respectively. Finally, conclusions and future research are mentioned in Sect. 8.

2 State of the art

The methods to assess railway capacity can be divided into different categories. One characteristic of a method is if it is timetable independent or timetable dependent. Examples of timetable-independent methods are presented by Jensen et al. (2017)

and Weik et al. (2020). The advantage of these type of methods is that they can be used in the strategic planning phase since no timetable is required yet. However, when the timetable is available, in a later stage of planning, it is more useful to use a timetable-dependent method because railway capacity depends heavily on how the infrastructure is used. Examples of factors that influence the capacity are the signalling system, train speeds, frequencies and routes. More relevant factors are discussed in Abril et al. (2008) and Besinovic and Goverde (2018). In this paper, we opt for a timetable-based method. Several (timetable-dependent) approaches are possible to assess the capacity.

Another classification of railway assessment methods can be made as follows: optimization models, simulation methods and analytical methods. An example of a timetable-based optimization model is given by Pellegrini et al. (2015). A recent simulation-based method is presented by Pascariu et al. (2021). However, in this paper we work with an analytical method because these methods are generally easy to use in practice. The most well-known analytical method is the timetable compression proposed by the International Union of Railways (UIC 2013). This method and its shortcomings are discussed in the following paragraphs.

One measure to quantify capacity is the capacity occupation. More generally, capacity occupation is one of the relevant metrics to determine the quality of a timetable (Gestrelius et al. 2021). The UIC defines 'capacity occupation' as the utilization of the infrastructure along a given section, measured over a defined time period (UIC 2013). In other words, the capacity occupation of a network is the shortest time that a sequence of trains needs to drive through that network. The leaflet UIC 406 proposes timetable compression as a method to determine the capacity occupation. In the first version of this leaflet (UIC 2004), this method decomposes the network into different parts and then considers the timetable during a specific time window on those network parts. The routes of the trains belonging to that timetable are shifted as close as possible to each other. The time that this compressed timetable then occupies the infrastructure is called the capacity occupation. For a cyclic timetable, this occupation time is given by the minimum cycle time.

Soon some criticisms on this method were formulated. Landex et al. (2006) argued that the capacity occupation of infrastructure elements such as line sections strongly depends on the length of the decomposed parts. This is in contrast to what would be expected from such a measure. However, no indication is given on how to decide on the appropriate length of the decomposed parts. Further, Lindner (2011) noticed that no recommendations were made about capacity occupation for nodes and stations, although they are clearly very important in capacity analysis of a network. In a revised version, the UIC made some adjustments for nodes (UIC 2013). Nodes are proposed to be divided in a platform area and one or more switch areas. However, this division leads to an underestimation of the real capacity occupation of the node or station because certain route dependencies are neglected (Bešinović and Goverde 2018).

Therefore, a new technique that is able to perform a timetable compression to a node as a whole (not decomposed in tracks and switches) was developed by Bešinović et al. (2017). The technique is based on the max-plus algebra and is called the max-plus automata model. It allows to consider all route dependencies between

trains by considering the node in microscopic detail. This method will be explained in detail in Sect. 4.1. An advantage of this method is that it only requires very simple mathematical operations and is therefore easy to use in practice.

Next to the criticism that nodes should not be decomposed to assess the capacity, it can also be argued that decomposing the network in lines and nodes does not lead to accurate results concerning the capacity. For example, it is possible that trains can follow each other very closely in a station area, but more spacing is needed between the trains when considering the adjacent corridor. This is exactly what Jensen et al. (2017) show: capacity occupation of a network is higher than the capacity occupation of the individual components of which the network consists. This means that a certain part of the infrastructure performs worse from a capacity perspective when considered as part of a network than when it is considered individually. Therefore, when quantifying the capacity of a network it is important to consider larger network zones to incorporate all train dependencies and network effects (Besinovic and Goverde 2018).

In conclusion, to accurately assess the railway capacity of a network it is necessary to consider the entire network in microscopic detail to include all the train dependencies. This summarizes the gap in current literature. The current approaches do not consider large networks as a whole with a microscopic level of detail. Methods based on the max-plus automata are suitable for small stations and simple corridors, with microscopic detail, but have not been applied to large networks yet. That is what will be done is this paper.

3 Definitions

In this section, we define the different concepts that will be used to analyse the capacity of railway networks.

3.1 Timetable compression

Timetable compression is the process of shifting trains as close as possible to each other in the time–space diagram. The resulting minimal time is defined as the *capacity occupation* (Bešinović and Goverde 2018). The capacity occupation measures the amount of time that a part of the infrastructure is occupied by a sequence of train operations. In this definition it is important to mention explicitly what is meant by "occupied by trains". This corresponds to "occupying" or "reserving" a part of the infrastructure, according to the well-known blocking time theory, as explained in the next section.

Compressing the timetable also allows to consider critical paths and resources, defined later in this section. These concepts will help to identify the true bottlenecks in a network.

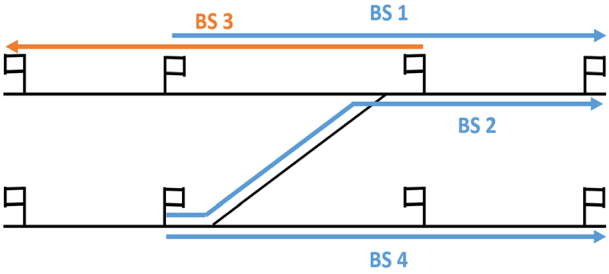

Fig. 1 Four block sections on a fictive part of the network infrastructure. BS 1, 2 and 4 range from and to stop signals for trains driving from left to right. BS 3 ranges from two stop signals in the opposite driving direction (right to left)

3.2 Blocking time theory: block sections and blocking time

To perform any (capacity) analysis on a train network, it is important to correctly calculate the necessary spacing between trains. This can be done using the blocking time theory (e.g. Bešinović and Goverde 2018).

In this theory, the railway network is typically divided into pieces of infrastructure that are exclusively allocated to a single train at a given time. These pieces are called block sections and go from a stop signal to the next stop signal in the same direction; see Fig. 1. Therefore, the direction of a block section is given by the direction of the stop signal at the beginning of this block section.

In some cases, for example when there is a switch, block sections can overlap. In this case, the considered block section is not only allocated to the train passing by, but also the overlapping block sections cannot be used by other trains. For example, in Fig. 1, if BS 2 is used, all other block sections are blocked indirectly. But if a train uses BS 1, another train can still use BS 4 and hence BS 4 is not blocked indirectly by a train using BS 1.

The blocking time of a block section is the time the block section is solely reserved by or allocated to one train and cannot be used by another. According to the blocking time theory, this consists not only of the running time through the block section, but it also includes the approaching time, the clearing time, etc. (e.g. Bešinović and Goverde 2018). The blocking times along a train's route form the blocking time stairway in the space–time diagram.

3.3 Resources

For the compression of the timetable, we need to model the trains on small pieces of infrastructure. Since working with block sections becomes difficult due to the many overlaps in switch areas, we opt for a new, smaller and unique piece of infrastructure called *resource*; see Fig. 2. A resource does not have a direction: it corresponds to a physical part of the track. A resource is chosen in a specific way to account for two things.

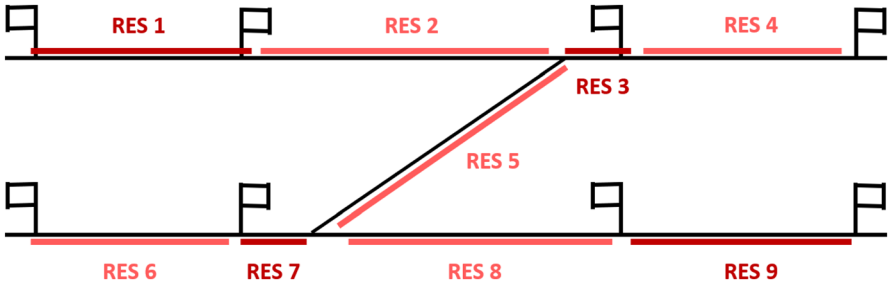

Fig. 2 Illustration of the definition of a resource on a fictive part of the network

Firstly, the blocking times of a train need to be projected correctly on the resources it passes. As blocking times are computed on block sections (from stop signal to stop signal), this means that a resource should be contained entirely in a block section. In other words, a block section can contain one or more resources (e.g. block section BS 2 in Fig. 1 contains resources RES 7, RES 5, RES 3 and RES 4 in Fig. 2), but a resource cannot lie on two consecutive block sections in a train route. Note that a resource can lie on two overlapping block sections, e.g. resource 4 (Fig. 2) lies on block sections BS 1 and BS 2 (Fig. 1). As a result, resources RES 7, RES 5, RES 3 and RES 4 will have the same blocking time, implied by BS 2.

Secondly, a resource should be well chosen concerning route possibilities. A resource should be small enough such that different routes that have nothing in common do not use the same resource. Otherwise, a train would not be allowed to continue its route when another train is using this resource, although both routes have no real infrastructure in common. This would result in an incorrect computation of capacity occupation. Resources must also be chosen in such a way that occupying one resource does not imply another resource to be unavailable indirectly. Either a resource is part of the train route and blocked directly, or the resource is not a part of the train route and it should not be blocked indirectly either. This can be taken care of by not including switches in a resource. A switch can hence be an end point of a resource, but not an internal point. The concept of a resource is related to the sets of basic infrastructure elements that Wakob introduces in his queuing model to analyse railway capacity. A set of basic infrastructure elements is constructed such that if one element of the set is occupied by a train, then all other infrastructure elements in the set are also blocked (De Kort et al. 1999). A resource can thus be seen as a set of basic infrastructure elements that is also small enough such that there are no elements in the set that can be blocked indirectly.

After performing a timetable compression, the *resource capacity occupation* can be obtained for all resources. This is simply defined as the end time of the last blocking time on each resource in the compressed timetable. The capacity occupation that was introduced earlier corresponds to the maximum resource capacity occupation.

3.4 Critical resources and paths

Critical resources and critical paths are concepts defined in the compressed timetable. In literature, a single critical path is defined for an entire compressed timetable. It starts from the last train in the compression, and thus the train that defines the capacity occupation, and consists of the chain of trains that touch each other after compression. However, since we are working on larger networks, it is interesting to extend this concept. Due to the many resources in the network, it is likely that the last train in the network only uses a small fraction of the resources. In that case, also the critical path possibly contains only a fraction of the entire set of critical resources in the network. This motivates us to introduce a set of so-called "trains-on-top" which contains all trains that are the final train on at least one resource. Each of the trains in this set will then give rise to a (partially) different critical path. A critical resource is then defined as a resource on which the blocking time stairways of two trains in a critical path touch each other. The critical paths and critical resources help to identify the true bottlenecks in a network. In case of a large network, having a set of critical paths instead of just one gives a better representation of the critical resources of that network. These concepts are now illustrated based on Fig. 3.

Figure 3 shows a compressed timetable of four trains. All trains in this example are "trains-on-top". The red train is on top for resources 1 and 9, the blue train for resources 3 and 6, the green train is on top for resources 4, 5, 7, 11 and 12 and finally the yellow train is on top for resources 2, 8 and 9. First, consider the critical path of the green train as an example. Since the green train touches the blue train on resource 4, this is a critical resource. Then search along the blue train for a next critical resource. This is resource 8 where the blue train touches the red train. For the red train, resources 1 and 2 are critical because they start at time 0. Now, look at the critical path of the yellow train. This train touches the blue train on resource 2. The remainder of the critical path is the same as for the green train. To summarize, the critical path of the green train contains the green, blue and red train and the critical resources are resource 1, 2, 4 and 8. For the yellow train, the critical path consists of

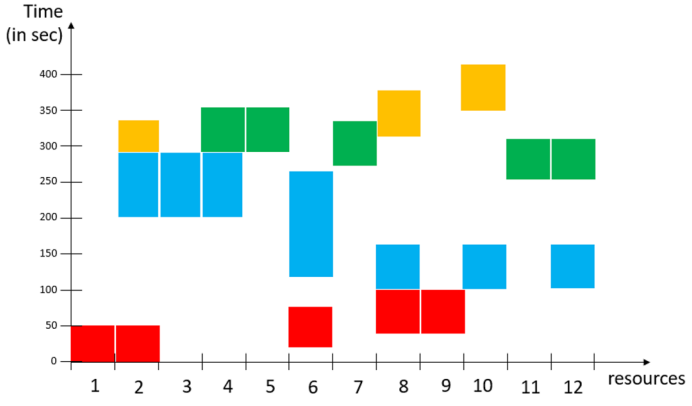

Fig. 3 An example of a compressed timetable with four trains, each colour represents a different train

the yellow, blue and red train. The critical resources are 1, 2 and 8. The yellow train also determines the maximum resource capacity occupation for this set of resources, namely on resource 10.

4 Methodology

We start by discussing the existing timetable compression method and the max-plus automata model to analyse capacity. Then, we explain the difficulties encountered when applying this method on a large and complex network and our adjustments to the compression method.

4.1 Timetable compression using max-plus automata model

Max-plus algebra

The max-plus algebra is an algebraic structure, called a semiring over the field $\mathbb{R}_{max} = \mathbb{R} \bigcup \{\varepsilon = -\infty\}$. It is equipped with two operations: maximum, denoted by $\oplus$ and addition (sum), denoted by $\otimes$. The definition of the operations goes as follows:

$$a \oplus b = \max(a, b),$$

$$a \otimes b = a + b,$$

for elements $a, b \in \mathbb{R}_{max}$.

If you take the maximum of an element and $-\infty$, the result is always the element. This means that $\varepsilon = -\infty$ is the *neutral element* for the operation $\oplus$. If you take the sum of an element and $-\infty$, the result is always $-\infty$. One says that $\varepsilon = -\infty$ is the *absorbing element* for $\otimes$. The neutral element for $\otimes$ is 0. In what follows, we will denote 0 by e.

The max-plus operations can also be extended to matrices. Let us denote the n by n matrices in the max-plus algebra with $\mathbb{R}_{max}^{n\times n}$. If $A = \left(a_{ij}\right)$ and $B = \left(b_{ij}\right) \in \mathbb{R}_{max}^{n\times n}$ are two matrices, then the matrix addition $\oplus$ (taking the maximum) and the matrix multiplication $\otimes$ (taking the maximum sum) are defined as follows.

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

$$[A \otimes B]_{ij} = \oplus_{k=1}^{n} a_{ik} \otimes b_{kj} = \max_{k=1,\ldots n} \left(a_{ik} + b_{kj}\right).$$

The first operation takes the maximum of every two matrix elements. The second operation considers the sum of every row element with the corresponding column element and then takes the maximum of these. More information on the max-plus algebra can be found in Baccelli et al. (1992), Heidergott et al. (2014) and Butkovic (2010).

Blocking time stairways in terms of the max-plus algebra

The blocking times of a train are given for each resource it uses in its route and can be visualized in the blocking time stairway. This blocking time stairway can then be represented by a max-plus matrix (Bešinović and Goverde 2018). This is a square matrix with dimensions equal to the number of resources in the considered network.

We denote the set of resources by R and the set of trains considered by T. Denote the number of resources in R by r. Then for every train $t \in T$ we call $M(t) \in \mathbb{R}_{max}^{r \times r}$ the *blocking time matrix* of train t, that is the matrix representing the blocking time stairway of t. Before we can define it properly, we need to introduce some other definitions.

Let $s_i(t)$ and $f_i(t)$ be the start and the end of the blocking time of train t on resource i and $R(t) \subset R$ the set of resources, used by t. Then the blocking time matrix $M(t)$ is defined as

$$M_{ij}(t) = \begin{cases} e & \text{if } i = j,\ i \notin R(t), \\ f_j(t) - s_i(t) & \text{if } i, j \in R(t), \\ \varepsilon & \text{else.} \end{cases}$$

In words, this matrix gives the difference between the end of the blocking time on one resource and the beginning of the blocking time on another resource (if both used by the train). On the diagonal of the matrix we hence have the blocking times for each resource. If not used, the diagonal contains $e = 0$ (the first line of the definition implies this). Elsewhere, the matrix contains $\varepsilon = -\infty$.

Timetable compression as max-plus matrix multiplication

Suppose we have two matrices representing the blocking time stairways of two trains. Multiplying these matrices visually corresponds to putting the blocking time stairways directly on each other in the graph. In other words, the blocks of the blocking time stairway are placed as low as possible in the graph, while still respecting the order of the trains in every resource. Indeed, if a resource is used by both trains, the block of the second train is always above the block of the first train. This is the case for every resource used by both trains. In short, we can say that the max-plus multiplication of two blocking time matrices corresponds visually to stacking their blocking time stairways onto each other. Therefore, the max-plus algebra allows to calculate efficiently the blocking time stairways of all trains after the compression.

Suppose it is possible to represent the blocking time stairways of the trains in the timetable as blocking time matrices and that the order in which the trains pass the network is given. Let n trains $t_i, i = 1 \dots n$, go through the network according to the order sequence $w = t_1 \cdots t_n$. Then the timetable compression, and hence the capacity occupation, can be calculated by multiplying the corresponding matrices one by one. In symbols,

$$M(w) = M\big(t_1 \cdots t_n\big) = M\big(t_1\big) \otimes \cdots \otimes M\big(t_n\big).$$

The matrix $M(w)$ can be seen as the representation of all blocking time stairways as close to each other as possible (i.e. compressed). The capacity occupation of each

resource is given by the maximum of the corresponding column in $M(w)$.We define the *upper contour* U of the compression as the row vector that contains the resource capacity occupation for each resource. It is calculated by multiplying upfront with a row vector of zeros $R(0)$:

$$U = R(0) \otimes M(w).$$

Remember that the maximum of all the resource capacity occupations in U is defined as the *capacity occupation* (Bešinović and Goverde 2018). This corresponds to the end time of the last blocking time in the compression.

It should be noted that all this is the case when we are talking about a general timetable, not necessarily cyclic. In case of a cyclic timetable, the timetable compression is performed including the first train(s) also after the last trains of the cycle. The capacity occupation is then computed not until the last block section of the last train, but just until the first block section of the new cycle. This can be lower than the last block section of the last train, if it is on another resource.

4.2 Adjustments for large and complex networks

Some difficulties arise when applying this method to a large and complex network on microscopic scale. In this section, we pinpoint these difficulties and offer solutions. Topics that will be discussed are determining the train sequence, splitting trains, keeping split trains together and speeding up the computation times.

4.3 Initial train sequence

To perform a compression based on a timetable, it is important that the order of the trains, according to the timetable, is preserved on the individual resources. When most trains visit most resources, determining this order is typically straightforward. However, in larger networks, this becomes more complicated. If a number of trains use a certain resource, it is important that the order of those trains on that resource is the same in the compressed timetable. This will help us to build a general order for the trains to perform the compression with. In fact, it is the only constraint that matters for building a train order: making sure that the implied order on resource level is still valid for the general train order.

How is this train order built? It starts with all the order relations amongst the trains. For example, train A and train B have resource X in common. On that resource, train A comes before train B. Then the order relation to take into account in building the general train order is that train A needs to come before train B.

From this reasoning, it also follows that two trains C and D that do not have a resource in common, also do not have a direct order relation. In terms of the order for the compression, this means that it does not matter if that train C is placed before train D or vice versa. As they do not have a resource in common, compressing train C first and then train D would yield the same result as compressing train D first and then train C.

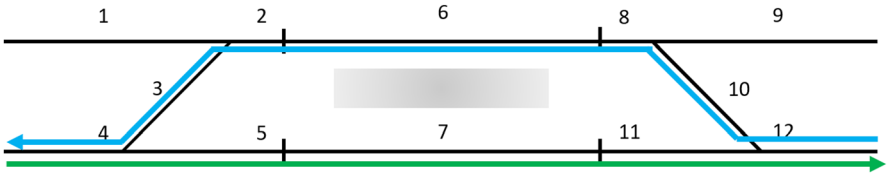

Fig. 4 Blue and green train driving on a fictive network with a station in the middle (colour figure online)

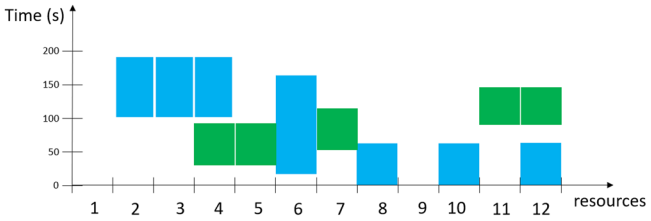

Fig. 5 Blocking time stairways of the blue and green train shown in Fig. 4 (colour figure online)

4.4 Train splitting

However, an important situation to keep in mind, when building a suitable train order, is the following. Consider two trains driving on a network as illustrated in Fig. 4. The corresponding timetable, illustrated by the blocking time stairways, is shown in Fig. 5. The blue train drives from right to left and stops on the upper platform of the station. The green train drives from left to right and passes at the lower platform. In this case, the blue train and the green train have multiple resources (4 and 12) in common. Moreover, on resource 4, on the left, the green train passes before the blue train and on resource 12, on the right, the blue train passes before the green train.

In this case, it cannot be concluded that the blue train comes before the green train because that does not hold for all of the resources. More specifically, it does not hold for resource 4. The other way around does not work either, because of resource 12. Thus, there is no order of these two trains that holds on all the resources of the network. Therefore, we propose to virtually split one of the trains, in order to be able to create a train order that preserves all order restrictions on the resources.

In the example, the blue train can be split into pieces (the green train would also work): its route until the station (dark blue) and its route from the station onwards (light blue). In the general train order, all resource orders can be satisfied by putting dark blue before green, which is followed by light blue. The correct compression of the timetable given in Fig. 5 is then shown in Fig. 6. Obviously, this situation can also occur for more than two trains. It is important to eliminate all these "order conflicts" by splitting certain trains on specific resources to construct a correct general train order. Finally, notice that the splitting of the trains is simply a tool to perform the compression correctly, there is no actual physical split of the trains imposed.

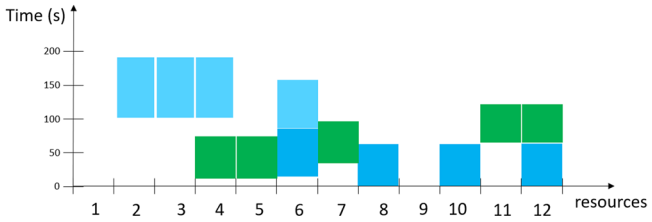

Fig. 6 Compressed timetable with split blue train. Train order: dark blue, green and light blue (colour figure online)

4.5 Keeping split parts of a train together

When splitting trains, another question arises. How to make sure that the two parts of the train stay together in the compression? In other words, how do we make sure that the first part is positioned correctly with respect to the second part?

A first simple adjustment is to add the blocking time of the resource where the train is split partly to the first train part and partly to the second train part. In that way, the second part will not descend more (or be further compressed) than where the first part ends. Of course, the first part should also not descend more than it should which would result in a gap between the correctly placed second part and the too low first part of the train. In that case, the first part of the train should be "pushed" upwards in the compression.

To do this, we introduce so-called "dummy resources". This type of resource is a virtual piece of infrastructure. The idea is that if the first part of a train needs to be pushed upwards x seconds, then we can add a dummy resource where there is a block of height x plus the time where the first block of the train currently begins. Including this resource in the compression method will guarantee that both parts of the train are still connected. This is illustrated in Fig. 7. This is an example of a compressed timetable that contains three trains: a green one, a yellow one and a blue one that has been split in dark and light blue. Notice that the blue train was split to solve the order conflict that occurs with the yellow train. If the dummy resource (on the right) is not included, the light blue part of the train would drop down until it touches the green train (on resource 2). Then there would be an unacceptable gap between the light blue and dark blue part of the train.

4.6 Reuse of trains

When one train has the same end station as another train's begin station, the same physical train is sometimes used to serve both train lines. This is called the reuse of a train. When a train is reused in a station, there is a minimal time between the arrival in the station and the departure in the other direction. To model this minimal time we also use a dummy resource. This resource will then be used by both trains to make sure that the second train cannot depart earlier than the minimal time after the first train.

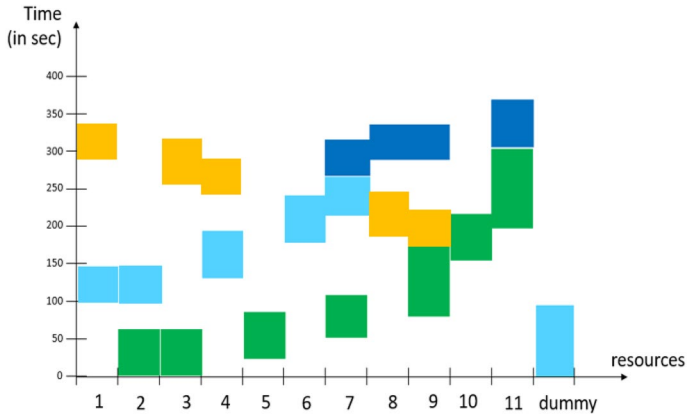

Fig. 7 Illustration of the use of a dummy resource to keep different parts of a single train together in the compression. The light and dark blue blocks are one train, the green blocks are a train and the yellow blocks are another train (colour figure online)

4.7 Faster computation times

The *original algorithm* for the timetable compression, see last subsection in Sect. 4.1., multiplies all blocking time matrices. These are matrices of dimension R^2, with R the number of resources in the network. As explained above, calculating the upper contour U_k in every step of the multiplication, allows us to determine the critical resources. This upper contour actually contains all the information we need to evaluate the capacity of the network. This approach corresponds to the state of the art described in literature.

One step in the *original algorithm* looks as follows. Let $M(t_k)$ be the blocking time matrix of train k and

$$M(t_1) \otimes \cdots \otimes M(t_k) = M(t_1 \cdots t_k)$$

the matrix representing the compression until train k (train k included). Then, in the next step, we multiply with the next blocking time matrix

$$M(t_1 \cdots t_k) \otimes M(t_{k+1}) = M(t_1 \cdots t_{k+1}).$$

After that, we compute the upper contour of the compression until train $k+1$ as follows:

$$R(0) \otimes M(t_1 \cdots t_{k+1}).$$

In the following step of the algorithm, we move on with the matrix $M(t_1 \cdots t_{k+1})$.

Our *new algorithm* starts with the row vector of zeros and then multiplies with all blocking time matrices one by one. In each step, the result of the multiplication is again a row vector, corresponding to the upper contour. This reduces the number of max-plus operations considerably.

The *new algorithm* looks as follows. We start with the zero contour $R(0)$, a row vector with zeros (dimension $1 \times R$). In the first step we multiply with the blocking time matrix of train 1

$$R(0) \otimes M\left(t_1\right) = R(0t_1).$$

This results in a new row matrix, which we will denote by $R\left(0t_1\right)$, and represents the upper contour of the compression until train 1 in this case. An arbitrary step in the algorithm is

$$R\left(0 \cdots t_k\right) \otimes M\left(t_{k+1}\right) = R\left(0t_1 \cdots t_{k+1}\right).$$

In each step, we immediately have the upper contour in this algorithm, so we do not need an extra multiplication with the zero-row matrix $R(0)$ (as in the *original algorithm*). Also, as we are multiplying a row vector with a matrix, we only have R max-plus operations, while the multiplication of two matrices implies R^2 max-plus operations. Both methods obtain the same result, but the new algorithm results in a significant gain in computation time. This allows to solve larger instances and also to use the timetable compression (or capacity occupation computation algorithm) in new ways, for example in an optimization problem.

4.8 Critical resources in the max-plus automata model

In this paragraph, it is explained how the critical resources can be computed directly from the max-plus multiplication. It can also be obtained from the visualization, but it is more efficient to directly compute it during the computation.

In the max-plus multiplication, the critical resources can be obtained in every multiplication step. Assume that the first k trains have been compressed. In order to determine the critical resource that determines the position of train $k + 1$ in the compression, the upper contour of the already compressed trains is required, denoted by U_k. By using the new algorithm described in the previous section, U_k is calculated in each step.

The critical resource can now be determined by multiplying U_k with $M\left(t_{k+1}\right)$. Notice that this simply corresponds to the next step in the new algorithm. Remember that each entry of a max-plus matrix multiplication is obtained by taking the maximum of some values. It follows from the definitions that the index for which this maximum is obtained corresponds to the critical resource. Note that this index must correspond to a resource that is used by the newly added train t_{k+1}.

5 Network instances

To demonstrate our method, we use a part of the Belgian railway network just outside of the main bottleneck in Brussels. In Sect. 5.1, the different zones of the network, considered in the case study, are introduced. Before performing an extensive case study, the performance and required computation times of the method are

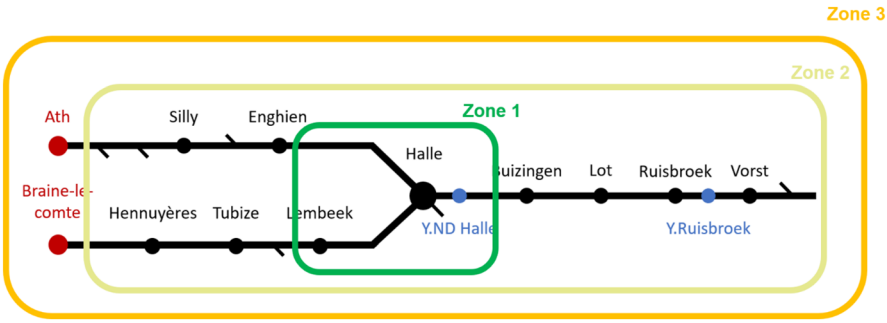

Fig. 8 Simplified macroscopic view of the studied network. The black dots are stations, the blue dots are complex switch areas and the red dots are considered as end stations in zone 3 (colour figure online)

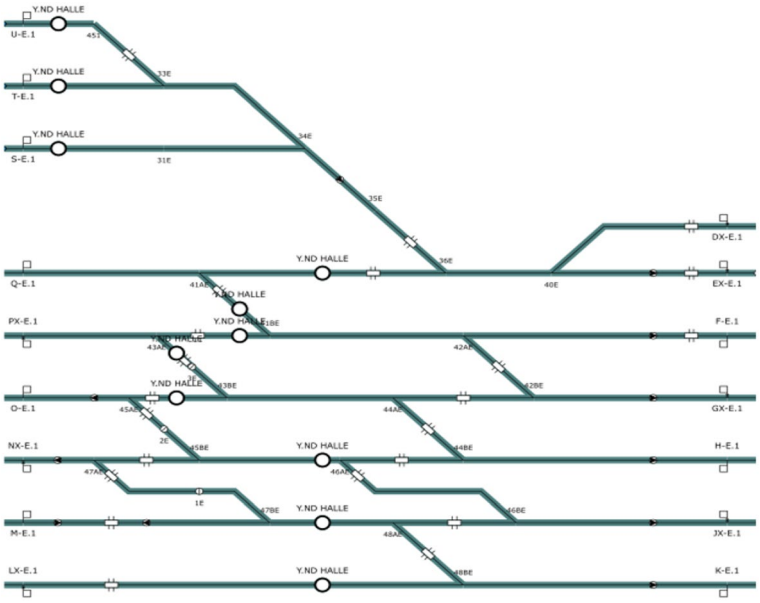

Fig. 9 Switch area Y.ND Halle

evaluated by considering the different zones with a different number of trains, as explained in Sect. 5.2.

5.1 Networks considered in the case study

In Fig. 8, the different zones of the network are shown. The three zones are centred around the main station of Halle, a medium-sized station with seven tracks and five platforms. Next to the main station is a large switch area Y.ND Halle. To illustrate

the complexity of the network, a microscopic view of this switch area is shown in Fig. 9. This switch area consists of 20 switches and at most 9 parallel tracks. The timetable during 1 h in the morning peak is considered.

Zone 1 consists of the area directly centred around the station of Halle. This zone contains 98 resources and 19 km of tracks. During the 1-h morning peak, 32 trains drive through the network.

Zone 2 centres around zone 1 and contains three additional directions. On the one side it ranges until Brussels-South (not included) and on the other side it goes to Ath (not included) and Braine-le-Comte (not included). It consists of 380 resources and 141 km of tracks. During 1 h of the morning peak, 37 trains pass through this zone. This consists of 15 local trains, 13 intercity trains, 6 peak hour trains and 3 high-speed trains.

Zone 3 consists of zone 2 with two additional stations: Braine-le-Comte and Ath. The network includes 585 resources and counts 177 km of tracks. During the 1-h morning peak, 49 trains drive through the network.

5.2 Reuse of a train

For zone 3, we take into account that two trains use the same physical train. So, at the station Braine-Le-Comte, when train TR24 arrives, the train has to wait (and "turn") on the platform and then train TR28 starts its route in the other direction. For these trains, we assume that there should be minimal 7 min between the arrival of the first train and the departure of the second train. In the timetable, there is currently a scheduled time between them of 10 min.

With this in mind, we also define a zone 2*. This is the same zone as zone 2, but we take the train turning of trains TR24 and TR28 (which happens outside the zone) into account. This means that we calculate the minimal time that is needed between these trains: we add the time that train TR24 reserves to drive outside zone 2 until the platform plus the time that train TR28 reserves from the platform until it enters zone 2. We add them both to the minimal reuse time that was used for zone 3 (which is in this case 7 min). This total time is then included on a dummy resource to extend zone 2 to zone 2*. In this way, we can evaluate the impact of including events that take place outside a considered zone in the timetable compression.

5.3 Network instances to test algorithm performance

As mentioned above, we will first experiment with a different number of trains in these zones to test the performance of the method and illustrate the difference in computation time. The three zones illustrated in Fig. 8 are used, but without taking the reuse of the trains into account. The different trains considered in each zone are determined as follows: start from zone 3 and randomly remove 10 of the 49 trains. The same trains are also removed in zone 1 and zone 2,

if they also run through these zones. This process is then repeated three more times. Table 1 summarizes which trains are used in the different instances.

Table 1 For each of the three zones there are five cases, A to E, with an increasing number of trains. The number of trains in each instance is indicated in bold. The train numbers are given such that all the trains used in a certain case can be found in the cell of that case and all cells to the left of it

	A	B	C	D	E
Zone 1	**5** = TR9, TR25, TR27, TR42, TR49	**12** = 5 + TR3, TR14, TR16, TR20, TR29, TR38, TR45	**18** = 12 + TR7, TR10, TR19, TR21, TR22, TR48	**26** = 18 + TR11, TR12, TR13, TR15, TR17, TR26, TR28, TR36	**32** = 26 + TR2, TR5, TR6, TR8, TR46, TR47
Zone 2	**7** = TR4, TR9, TR25, TR27, TR37, TR42, TR49	**14** = 7 + TR3, TR14, TR16, TR20, TR29, TR38, TR45	**20** = 14 + TR7, TR10, TR19, TR21, TR22, TR48	**28** = 20 + TR11, TR12, TR13, TR15, TR17, TR26, TR28, TR36	**37** = 28 + TR1, TR2, TR5, TR6, TR8, TR23, TR24, TR46, TR47
Zone 3	**9** = TR4, TR9, TR18, TR25, TR27, TR32, TR37, TR42, TR49	**19** = 9 + TR3, TR14, TR16, TR20, TR29, TR30, TR35, TR38, TR41, TR45	**29** = 19 + TR7, TR10, TR19, TR21, TR22, TR31, TR33, TR34, TR44, TR48	**39** = 29 + TR11, TR12, TR13, TR15, TR17, TR26, TR28, TR36, TR40, TR43	**49** = 39 + TR1, TR2, TR5, TR6, TR8, TR23, TR24, TR39, TR46, TR47

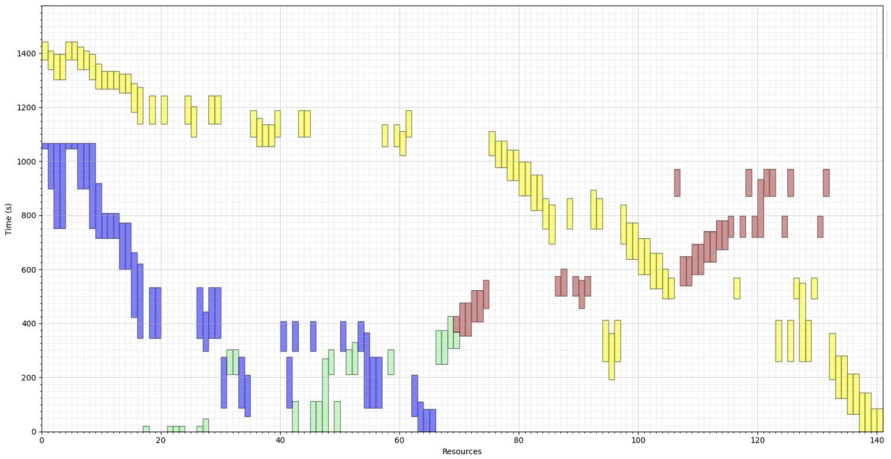

Fig. 10 Loop of three trains results in four trains TR3-1 (green), TR22 (blue), TR36 (yellow) and TR3-2 (red) in the compressed timetable (colour figure online)

6 Experimental results

In this section, the experimental results are presented. First, the train orders are determined for the three zones, and train splits are used for this. The following sections give the results about the algorithm performance and the case study. A regular laptop with an Intel Core i5-4210U CPU @ 1.70 GHz 2.40 GHz processor, 16 GB RAM was used to run the code.

6.1 Train orders and train splits

To perform the compression method on the different zones, the train order needs to be determined. This is straightforward for zone 1, no train splits are required in that case. The resulting train order can be found in Table 6 in the Appendix. For zones 2 and 3, train splits are necessary to obtain a correct order.

In zone 2, two order conflicts or so-called loops occurred: a loop of three trains (TR3, TR22, TR36) and a loop of five trains (TR15, TR8, TR5, TR38, TR7).

The loop of three trains was solved by splitting train TR3 (in TR3-1 and TR3-2). For those trains, this results in the train order TR3-1, TR22, TR36, TR3-2. These four trains in the compressed timetable are visualized in Fig. 10. We can clearly see that the two train parts TR3-1 (green) and TR3-2 (red) are kept together in the compressed timetable, with one common resource (70 in Fig. 10).

The loop of five trains is also solved by just splitting one train, train TR7 into TR7-1 and TR7-2. The resulting order for the loop is TR7-1, TR15, TR8, TR5, TR38 and TR7-2. This results in a train sequence of 39 trains (the 37 trains and two additional trains due to the splits). The train sequence resulting from the order relations on the individual resources is given in Table 7 in the Appendix.

Table 2 Results for zone 1

	Capacity occupation (%)	Computation time new	Computation time original
A	16	≤0.1	0.2
B	33	≤0.1	0.4
C	42	≤0.1	0.6
D	50	≤0.1	0.8
E	58	≤0.1	1

Table 3 Results for zone 2

	Capacity occupation (%)	Computation time new	Computation time original
A	51	≤0.1	12.0
B	62	0.1	25.8
C	72	0.2	38.1
D	82	0.2	58.6
E	82	0.3	76.0

In zone 3, eight splits are needed to solve five loops. However, the number of loops is not entirely fixed as some new loops are detected while fixing the others. With eight splits, we have a total of 57 trains (49 original trains and eight extra trains due to splits). The train order that we will use in the max-plus automata model is given in Table 8 in the Appendix. This order satisfies all order constraints on resource level between the trains. It might be possible to solve these five loops by splitting less than eight trains, but the only aim here is to obtain a feasible train order. It should be noted that the way the loops are solved has no impact on the capacity occupation. With the train orders determined for the different zones, the timetable can be compressed.

6.2 Algorithm performance

In this section, the performance of the algorithm is analysed by applying the algorithm to the fifteen cases described in Table 1. The capacity occupation is reported in per cent and the computation time of both the original and the new algorithm to calculate the compression in seconds, rounded to one decimal. Both algorithms give the same result for the capacity occupation, they only differ in computation time. Tables 2, 3 and 4 give the results for zone 1, 2 and 3, respectively.

For each zone, the computation times show linear growth for both the original (Sect. 4.1) and the new algorithm (Sect. 4.2). This is a logical result, since adding more trains means doing more multiplications. As expected, the original method is

Table 4 Results for zone 3

	Capacity occupation	Computation time new	Computation time original
A	63%	0.1	61.3
B	95%	0.3	153.1
C	95%	0.5	228.1
D	95%	0.7	336.8
E	95%	0.9	431.0

considerably slower than the new method. For zone 1, the new method is on average 25 times faster than the original algorithm. This average factor becomes around 230 for zone 2 and 450 for zone 3. Thus, as the network becomes larger, resulting in more resources, the gain in computation time by using the new method instead of the original one grows. For the largest considered network, zone 3 case E, the computation time with the new method is around 1 s, while the original method takes over 7 min.

Some interesting observations can also be made about the capacity occupation. For zone 1, the capacity occupation increases quite evenly when more trains are considered. The same can initially be seen for zone 2, but the maximal value for the capacity occupation is already obtained in case D with 37 trains. For zone 3, the maximum capacity occupation of 95% is already obtained in case B where only nineteen trains were considered. To gain more insight into the results for zone 3, we consider the critical path of the train that determines the capacity occupation for instance E. This critical path only contains three trains: TR29, TR42 and TR45. The result is shown on Fig. 11. From Table 1, we see that these three trains are included in case B for zone 3 as well. This explains why the capacity occupation is the same for all the subsequent cases. Obviously, this also raises the question about the meaning or usefulness of the capacity occupation concept. The result of case B seems to imply that the network is almost "full", or at capacity, but 30 more trains can actually be added without an increase in capacity occupation. Another observation is that the three trains of the critical path that determine the capacity occupation are included in case B for zone 1 as well. The capacity occupation in that case is only 33%. This shows that enlarging the considered network can have a large impact on the capacity occupation.

7 Results of the case study

In this section we discuss the results obtained for the case study in terms of capacity occupation and critical paths and resources.

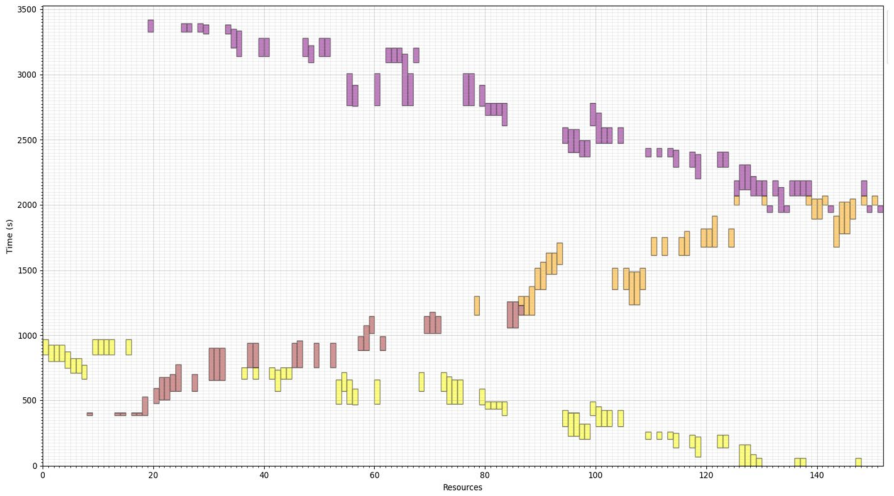

Fig. 11 Critical path of cases B, C, D and E in zone 3. The purple train is TR42, the brown and orange trains are TR29 (they were split to perform the compression) and the yellow train is TR45 (colour figure online)

Table 5 Capacity occupation for different zones

	Zone 1	Zone 2	Zone 2*	Zone 3
Capacity occupation (s)	2084	2936	3237	3420
Occupation rate	58%	82%	90%	95%

7.1 Capacity occupation

The results for the different zones are presented in Table 5. Notice that some of these results can also be found in Tables 2, 3 and 4 as instance E for each of the three zones. In Sect. 5.1, the reuse of a physical train in zone 3 was introduced. To include this constraint in zone 2 as well, zone 2* was introduced. As can be seen in Table 5, the capacity occupation for this zone is equal to 90%. This shows that incorporating a known dependency outside of the network gives a better estimation of the capacity occupation. However, the capacity occupation of zone 3 is still larger than that of zone 2*, thus there are other dependencies that are only considered when the network is extended. This confirms the general conclusion that in order to determine the capacity occupation of a network, the entire network should be considered during compression. Finally, recall that the capacity occupations presented here are obtained without incorporating the periodicity of the timetable, which could make the occupation lower, as explained at the end of Sect. 4.1.

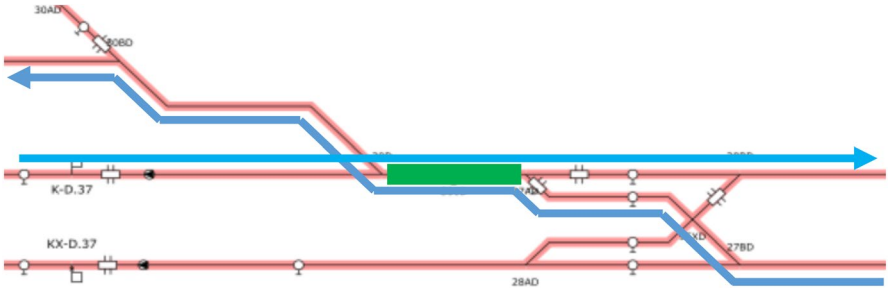

Fig. 12 Detail of resource 27AD-29D (green) in Enghien with the routes of trains TR7 and TR3 (blue arrows) (colour figure online)

7.2 Critical paths and critical resources

The critical resources of zones 2 and 3 are discussed, with and without incorporating the reuse of the train from train TR24 to train TR28.

In zone 2, the critical path leading to the reported capacity occupation contains the trains TR42, TR21, TR28, TR7 and TR3. It was stated in Sect. 3.4 that a critical path can be defined for each "train-on-top". In this zone, TR21, also included in the critical path for TR42, is also a train-on-top for other resources. There are also other trains-on-top that are not contained in this critical path. In total, there are actually 23 "trains-on-top". When considering only the ones that are on top for at least ten resources, 13 trains remain. In 11 of the 13 critical paths defined by the most common "trains-on-top", TR3 and TR7 occur.

The critical resource for these two trains is resource 27AD-29D in Enghien, where train TR7 follows train TR3; see Fig. 12. This resource just lies at the place where track L123 splits from track L94. Moreover, track L123 first has a part where trains in two directions need to pass on a single track and especially at resource 27AD-29D where trains in two directions have to pass. This explains why this specific resource pops up as a recurring critical resource.

On the other hand, when looking at the critical paths of zone 2*, i.e. with the reuse of the train, a clear change in recurring critical points is observed. Instead of the previous critical point, now the point where train TR24 follows train TR28 on the dummy resource occurs frequently. So, if train TR24 is forced to start later due to the minimal time for turning the train to reuse it, this results in a critical point and hence plays an important role in the capacity occupation of the network. This confirms the results shown in Table 5 that reusing the physical train of TR24 for TR28 increases the capacity occupation. Looking at this from another perspective, it can be concluded that, using a different train for TR24 and TR28 could reduce the capacity occupation and show other critical points in the network, such as the point in Fig. 12.

In zone 3, the critical path leading to the reported capacity occupation contains the trains TR42, TR29 and TR45. Thus, the train-on-top that determines the capacity occupation is the same for both zone 2 and zone 3, but the corresponding critical

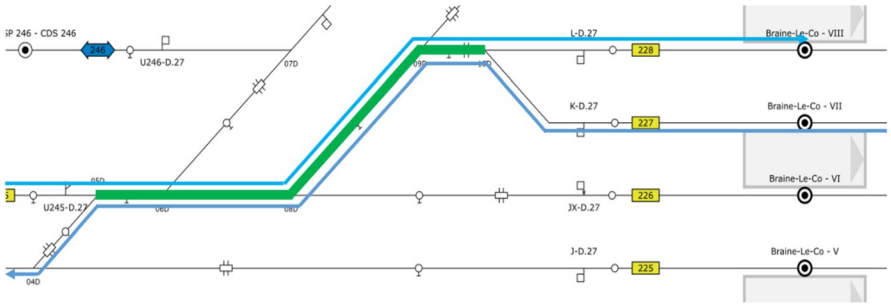

Fig. 13 Detail of resources 05D-06D, 06D-08D, 08D-09D and 09D-10D in Braine-le-Comte with the routes of trains TR29 and TR42 (blue arrows) (colour figure online)

paths are different. In total, there are 36 trains-on-top in zone 3, 12 of them are on top for at least ten resources.

Furthermore, another critical point occurs, beside the previous one for zone 2. This is between train TR29 and train TR42 on resources 05D-06D, 06D-08D, 08D-09D and 09D-10D in Braine-le-Comte; see Fig. 13. It corresponds to the route to platforms 7 and 8. It make sense that this point only occurs in critical paths in zone 3 as the resource is located in the station of Braine-le-Comte, which is only a part of zone 3 and not of zone 2. Alternative train routes or additional infrastructure could alleviate the use of this resource and reduce the capacity occupation in this zone.

8 Conclusions

We can conclude that the application of the timetable compression on a large and complex network is not straightforward. Important adjustments were required: determining an appropriate train sequence, splitting the trains due to train order conflicts, introducing dummy resources to keep the split trains together in the compressed timetable and to represent the reuse of trains, speeding up the algorithm and computing the critical path directly through the max-plus multiplication. These adjustments allow us to perform a timetable compression of larger and more complex networks than the ones previously considered in literature.

Some interesting observations were made during the different experiments. For example, the capacity occupation of zone 3 with nineteen trains is equal to the capacity occupation of the same zone with 49 trains. Actually, only considering the three trains of the critical path that determines the capacity occupation would also give the same result of 95%. The results also showed that the capacity occupation heavily depends on the size of the network that is considered. These observations indicate that it is not easy to give a clear, practical interpretation of the capacity occupation. It is, for example, not clear how this value can be used to say something about the possibility of adding another train to the network. For reliable results, the entire route of a newly proposed train should be contained in the considered network. Nevertheless, the method allowed to determine a number of critical paths and,

even more importantly, a number of critical resources in the zones considered. If additional trains are considered, they should avoid these critical resources as much as possible. When infrastructure extensions are considered, these are probably most efficient if they can alleviate the use of these critical resources.

The experiments showed a clear improvement in computation time when the new algorithm is used. In general, the computation times increase linearly when more trains were considered. The computational gain of the new algorithm becomes larger when the number of resources in the considered network increases.

In further research, it would be interesting to extend this work to cyclic timetables, in order to better take into account periodicity. Secondly, as it appears that larger networks have a larger capacity occupation than their constituting parts individually, it is also interesting to further investigate network effects that can account for these discrepancies. Lastly, it can also be worth using a mathematical model such as for basic production planning to compute the capacity occupation, which could also allow more possibilities for optimization problems regarding railway capacity.

Appendix 1

See Tables 6, 7 and 8.

Table 6 Train order for zone 1

1. TR3	9. TR36	17. TR5	25. TR46
2. TR10	10. TR45	18. TR6	26. TR25
3. TR27	11. TR8	19. TR14	27. TR42
4. TR47	12. TR19	20. TR9	28. TR26
5. TR20	13. TR29	21. TR17	29. TR13
6. TR22	14. TR7	22. TR28	
7. TR49	15. TR16	23. TR2	
8. TR15	16. TR38	24. TR21	

Table 7 Train order for zone 2 with train splits

1. TR1	11. TR45	21. TR6	31. TR25
2. TR24	12. TR3-2	22. TR17	32. TR42
3. TR48	13. TR19	23. TR38	33. TR49
4. TR10	14. TR29	24. TR7-2	34. TR4
5. TR20	15. TR16	25. TR14	35. TR11
6. TR12	16. TR47	26. TR28	36. TR26
7. TR3-1	17. TR7-1	27. TR9	37. TR13
8. TR22	18. TR15	28. TR21	38. TR23
9. TR27	19. TR8	29. TR46	39. TR37
10. TR36	20. TR5	30. TR2	

Table 8 Train order zone 3, with split trains

1. TR1	16. TR12-1	31. TR15-2	46. TR42
2. TR24	17. TR3-2	32. TR14	47. TR6
3. TR32	18. TR27-1	33. TR28	48. TR44
4. TR34	19. TR45	34. TR29-2	49. TR17-1
5. TR39	20. TR15-1	35. TR9	50. TR46-2
6. TR40	21. TR29-1	36. TR2	51. TR25
7. TR48	22. TR16	37. TR41	52. TR49
8. TR10	23. TR27-2	38. TR12-2	53. TR4
9. TR35	24. TR43	39. TR11	54. TR26
10. TR20	25. TR47	40. TR8-2	55. TR13
11. TR3-1	26. TR8-1	41. TR37	56. TR23
12. TR22	27. TR5	42. TR21	57. TR17-2
13. TR36	28. TR30	43. TR33	
14. TR19	29. TR38	44. TR46-1	
15. TR18	30. TR7	45. TR31	

Acknowledgements The authors would like to thank Infrabel, the Belgian railway infrastructure manager, and NMBS, the Belgian railway passenger operator for their collaboration and support.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Abril M, Barber F, Ingolotti L, Salido MA, Tormos P, Lova A (2008) An assessment of railway capacity. Transport Res Part E 44(5):774–806

Baccelli F, Cohen G, Olsder GJ, Quadrat JP (1992) Synchronization and linearity: an algebra for discrete event systems, New York, Wiley.

Bešinović N, Goverde RMP (2018) Capacity Assessment in Railway Networks. In: Borndörfer R, Klug T, Lamorgese L, Mannino C, Reuther M, Schlechte T (Eds) Handbook of Optimization in the railway Industry. International Series in Operations Research & Management Science, vol. 26. Springer, Cham, pp 25–45 https://doi.org/10.1007/978-3-319-72153-8_2

Bešinović N, Goverde RMP, Quaglietta E (2017) Microscopic models and network transformations for automated railway traffic planning. Comput-Aided Civil Infrastruct Eng 32(2):89–106. https://doi.org/10.1111/mice.12207

Butkovič P (2010) Max-linear systems: theory and algorithms. Springer Science & Business Media.

De Kort AF, Heidergott B, Van Egmond RJ, Hooghiemstra G (1999) Train movement analysis at railway stations: Procedures & evaluation of Wakob's Approach. TRAIL Studies in Transportation Science, S99/1, Delft University Press.

Gestrelius S, Häll CH, Peterson A (2021) Capacity utilization, travel time, stability and heterogeneity—a linear programming analysis for railway timetabling. In Proceedings of the RailBeijing Conference, Beijing, China, 3–7 November 2021.

Heidergott B, Olsder GJ, Van Der Woude J (2014) Max plus at work. In Max Plus at Work. Princeton University Press

Jensen LW, Landex A, Nielsen OA, Kroon LG, Schmidt M (2017) Strategic assessment of capacity consumption in railway networks: framework and model. Transport Res Part C 74:126–149. https://doi.org/10.1016/j.trc.2016.10.013

Landex A, Kaas AH, Schittenhelm B, Schneider-Tilli J (2006) Practical use of the UIC 406 capacity leaflet by including timetable tools in the investigations. WIT Transact Built Environ 88:643–652. https://doi.org/10.2495/CR060631

Lindner T (2011) Applicability of the analytical UIC Code 406 compression method for evaluating line and station capacity. J Rail Transport Planning Manag 1(1):49–57. https://doi.org/10.1016/j.jrtpm.2011.09.002

Pascariu B, Coviello N, D'Ariano A (2021) Railway freight node capacity evaluation: a timetable-saturation approach and its application to the Novara freight terminal. Transport Res Proc 52:155–162

Pellegrini P, Marlière G, Pesenti R, Rodriguez J (2015) RECIFE-MILP: an effective MILP-based heuristic for the real-time railway traffic management problem. IEEE Trans Intell Transp Syst 16(5):2609–2619

UIC (2004) Capacity (UIC Code 406) (1st edition).

UIC (2013) Capacity (UIC Code 406) (Issue 2nd edition).

Weik N, Warg J, Johansson I, Bohlin M, Nießen N (2020) Extending UIC 406-based capacity analysis–New approaches for railway nodes and network effects. J Rail Transport Plann Manag 15:100199