Kistosil Fahim Subiono; Fahim Kistosil; Dieky Adzkiya

Generalized public transportation scheduling using max-plus algebra

Kybernetika, Vol. 54 (2018), No. 2, 243–267

Persistent URL: `http://dml.cz/dmlcz/147192`

KYBERNETIKA — VOLUME *54* (2018), NUMBER 2, PAGES 243–267

GENERALIZED PUBLIC TRANSPORTATION SCHEDULING USING MAX-PLUS ALGEBRA

Subiono, Kistosil Fahim and Dieky Adzkiya

In this paper, we discuss the scheduling of a wide class of transportation systems. In particular, we derive an algorithm to generate a regular schedule by using max-plus algebra. Inputs of this algorithm are a graph representing the road network of public transportation systems and the number of public vehicles in each route. The graph has to be strongly connected, which means there is a path from any vertex to every vertex. Let us remark that the algorithm is general in the sense that we can allocate any number of vehicles in each route. The algorithm itself consists of two main steps. In the first step, we use a novel procedure to construct the model. Then in the second step, we compute a regular schedule by using the power algorithm. We describe our proposed framework for an example.

Keywords: max-plus algebra, strongly connected road network, scheduling

Classification: 15A15, 15F10

1. INTRODUCTION

The increase of population in a large city needs good and reliable transportation systems. Such transportation systems are characterized by the existence of an assurance to the passengers in the form of a regular schedule. The construction of a schedule for transportation systems has been discussed in the literature, for example in [4, 10, 18, 20]. There are many techniques to generate a schedule. One of them is using mathematical models. In the literature, there are many models of transportation systems [1, 5, 7, 8, 9, 16, 17, 18]. Max-plus algebra [2, 15] is a modeling framework that can be used to design a schedule for a transportation system. In this work, we construct a regular schedule for a wide class of transportation systems by using max-plus algebra. The regular schedule is constructed by using the spectral properties over max-plus algebra. Let us remark that although max-times algebra and max-plus algebra are isomorphic, it is not straightforward to leverage the spectral properties over max-times algebra discussed in [14] to spectral properties over max-plus algebra.

Some works on the scheduling of some systems using max-plus algebra have been discussed in the literature. In particular, the design of timetable of the train system in the Netherlands has been studied in [3]. The model only considers the case of intercity trains. Thus the corresponding model has a dimension of moderate size. As a follow-up,

DOI: 10.14736/kyb-2018-2-0243

the whole Dutch railway system with *all* train types has been investigated in [22]. The thesis [13] discussed the analysis of railway timetables. Besides that, [11] has addressed scheduling problem of monorail and tram in Surabaya city of East Java province. Finally, the scheduling of supply chain systems has been discussed in [23].

The procedure to construct a regular schedule using max-plus algebra requires a model in the form of Max-Plus-Linear (MPL) systems. In general, constructing such model is complicated and time-consuming. The complexity increases when the initial number of vehicles in each route is different. By using techniques in the literature [2, 15], we can construct the model in two steps: develop a higher-order MPL system and transform the model into a first-order MPL system. However, when we transform the higher-order MPL system into a first-order MPL system, we introduce some state variables when there exists a route containing more than one vehicles. As a consequence, the schedule generated from such model has more information than we need. Choosing the right information is a difficult task, especially when we work on a large-scale transportation network. The previous problem motivates us to propose a systematic framework to construct a regular schedule for any public transportation network by using max-plus algebra. The framework allows us to synthesize a regular schedule from a large-scale transportation network because the procedure can be implemented as a software tool.

Our framework requires that the graph of public transportation networks is strongly connected. This means we can travel from any place to every destination. Thus this requirement is not restrictive. Our framework consists of three main steps. In the first step, we construct an implicit higher-order MPL system from a public transportation network. Then we show that the obtained model has a unique max-plus eigenvalue. Then in the second step, we transform the implicit higher-order MPL system into an explicit first-order MPL system using a novel approach. The new approach allows us to design a systematic procedure to reduce the obtained model. To the best of our knowledge, by using the standard approach [2, 15], we cannot design such systematic procedure. Finally, in the third step, we determine the regular schedule by using the power algorithm [21]. In each step, we provide detailed explanations in order to convince the reader that those steps are systematic. As such, we can implement those steps as a computer program.

The paper is structured as follows. Section 2 discusses the models and some related notions. Then Section 3 describes the contributions of this paper. The main contribution is the procedure to construct a regular schedule in Section 3.3. Each step in the procedure is based on the discussions in Sections 3.1 and 3.2. In Section 3.1, we construct the model of public transportation systems and analyze its properties. Then, we reduce the size of the model in Section 3.2. The procedure is then illustrated on a case study in Section 3.4. Finally, Section 4 concludes the paper.

2. MODELS AND PRELIMINARIES

In this section, we briefly introduce max-plus algebra and some related notions [2, 15]. Furthermore, we also describe synchronization features in public transportation scheduling [11].

2.1. Max-plus algebra

We define $\mathbb{R}$ and $\mathbb{N}$ as the set of real numbers and natural numbers $\{1, 2, \dots\}$, respectively. The max-plus algebra is defined as $\mathbb{R}_{\max} = (\mathbb{R}_\varepsilon, \oplus, \otimes)$, where $\mathbb{R}_\varepsilon \overset{\text{def}}{=} \mathbb{R} \cup \{\varepsilon\}$ and $\varepsilon \overset{\text{def}}{=} -\infty$. The binary operators $\oplus$ and $\otimes$ are defined as follows:

$$x \oplus y \overset{\text{def}}{=} \max\{x, y\} \qquad \text{and} \qquad x \otimes y \overset{\text{def}}{=} x + y$$

for every $x, y \in \mathbb{R}_\varepsilon$. Thus in max-plus algebra, the addition and multiplication operations are replaced by maximization and the usual addition operation. Notice that ε is the neutral element of $\oplus$. The neutral element of $\otimes$ is defined as $e \overset{\text{def}}{=} 0$. Furthermore, in the context of max-plus algebra, $a^{\otimes b} = b \times a$, where × is the conventional multiplication operator. In this paper, the conventional multiplication operator × is usually omitted, whereas the max-plus multiplication operator $\otimes$ is always written explicitly. Let us remark that in max-plus algebra, every element $x \in \mathbb{R}$ has an inverse under the $\otimes$ operation, denoted by $-x$.

Next, we introduce matrices over $\mathbb{R}_{\max}$. The set of matrices of size $m \times n$ over the max-plus algebra is denoted by $\mathbb{R}_\varepsilon^{m \times n}$. For $n \in \mathbb{N}$, we define $\underline{n} \overset{\text{def}}{=} \{1, 2, \dots, n\}$. An element in the ith row and jth column of matrix $A \in \mathbb{R}_\varepsilon^{m \times n}$ is denoted by $[A]_{i,j}$ for $i \in \underline{m}$ and $j \in \underline{n}$. The max-plus identity matrix of size $n \times n$ is denoted by E_n, i. e. the elements on the main diagonal of the matrix are equal to e and the other elements are equal to ε. A max-plus zero matrix of size $m \times n$ is denoted by $\mathcal{E}_{m,n}$, i. e. all elements of the matrix are equal to ε.

Let $A, B \in \mathbb{R}_\varepsilon^{m \times n}$, the max-plus addition $A \oplus B$ is defined by $[A \oplus B]_{i,j} \overset{\text{def}}{=} [A]_{i,j} \oplus [B]_{i,j} = \max\{[A]_{i,j}, [B]_{i,j}\}$, for $i \in \underline{n}$ and $j \in \underline{m}$. For matrix $A \in \mathbb{R}_\varepsilon^{m \times n}$ and scalar $\alpha \in \mathbb{R}_\varepsilon$, max-plus scalar multiplication $\alpha \otimes A$ is defined by $[\alpha \otimes A]_{i,j} \overset{\text{def}}{=} \alpha \otimes [A]_{i,j}$, for $i \in \underline{n}$ and $j \in \underline{m}$.

For matrices $A \in \mathbb{R}_\varepsilon^{m \times p}$ and $B \in \mathbb{R}_\varepsilon^{p \times n}$, max-plus multiplication $A \otimes B$ is defined by $[A \otimes B]_{i,j} \overset{\text{def}}{=} \bigoplus_{k=1}^{p} [A]_{i,k} \otimes [B]_{k,j} = \max_{k \in \underline{p}}\{[A]_{i,k} + [B]_{k,j}\}$, for $i \in \underline{m}$ and $j \in \underline{n}$. For matrix $A \in \mathbb{R}_\varepsilon^{n \times n}$ and positive integer k, the kth max-plus power of A is denoted by $A^{\otimes k}$ and defined as $A^{\otimes k} \overset{\text{def}}{=} \underbrace{A \otimes A \otimes \cdots \otimes A}_{k \text{ times}}$. Similar with the conventional algebra, we have $A^{\otimes 0} \overset{\text{def}}{=} E_n$, for $A \in \mathbb{R}_\varepsilon^{n \times n}$. For ease of notations, we introduce $A^+ \overset{\text{def}}{=} \bigoplus_{k=1}^{\infty} A^{\otimes k}$ and $A^* \overset{\text{def}}{=} E_n \oplus A^+ = \bigoplus_{k=0}^{\infty} A^{\otimes k}$, where $A \in \mathbb{R}_\varepsilon^{n \times n}$.

2.2. Max-plus-linear systems

An autonomous Max-Plus-Linear (MPL) system that characterizes the dynamics of autonomous timed event graphs is given by [2, Th. 2.58]:

$$\mathbf{x}(k+1) = A_0 \otimes \mathbf{x}(k+1) \oplus A_1 \otimes \mathbf{x}(k) \oplus \cdots \oplus A_l \otimes \mathbf{x}(k+1-l), \tag{1}$$

where $A_s \in \mathbb{R}_\varepsilon^{n \times n}$, $\mathbf{x}(k+1-s) = [x_1(k+1-s) \quad \dots \quad x_n(k+1-s)]^T \in \mathbb{R}^n$ for $k \in \mathbb{N}$ and $s \in \{0, 1, \dots, l\}$. Vectors are written using the bold typeset, whereas the entries are denoted by the normal typeset with the same name and index, e. g. the ith component of vector $\mathbf{x}(k)$ is denoted by $x_i(k)$. The independent variable k denotes an increasing occurrence index, whereas the state variable $\mathbf{x}(k)$ defines the (continuous) time of kth occurrence of all events. In particular, the state component $x_i(k)$ denotes the (continuous) time of kth occurrence of ith event. Autonomous MPL systems are characterized by deterministic dynamics, namely they are unaffected by exogenous inputs in the form of control signals or of environmental non-determinism.

If $A_0 \neq \mathcal{E}_{n,n}$, the MPL system is called implicit. Otherwise if $A_0 = \mathcal{E}_{n,n}$, the MPL system is called explicit. An implicit MPL system (1) can be transformed into an explicit MPL system (2) under some condition (cf. Proposition 2.1). If the condition in Proposition 2.1 is satisfied, A_0^* exists and the explicit MPL system is given by

$$\mathbf{x}(k+1) = A_0^* \otimes A_1 \otimes \mathbf{x}(k) \oplus \cdots \oplus A_0^* \otimes A_l \otimes \mathbf{x}(k+1-l). \tag{2}$$

MPL systems of the forms (1) and (2) are called of order l. If $l > 1$, the MPL systems are called higher order, whereas if $l = 1$, the MPL systems are called first order. An explicit higher-order MPL system (2) can be transformed into the following explicit first-order autonomous MPL system [2, Rem. 2.75]:

$$\tilde{\mathbf{x}}(k+1) = \tilde{A} \otimes \tilde{\mathbf{x}}(k), \tag{3}$$

where $\tilde{\mathbf{x}}(k) = [\mathbf{x}(k)^T \quad \ldots \quad \mathbf{x}(k+1-l)^T]^T \in \mathbb{R}^{nl}$ and

$$\tilde{A} = \begin{bmatrix} A_0^* \otimes A_1 & \ldots & A_0^* \otimes A_{l-2} & A_0^* \otimes A_{l-1} & A_0^* \otimes A_l \\ E_n & \ldots & \mathcal{E}_{n,n} & \mathcal{E}_{n,n} & \mathcal{E}_{n,n} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathcal{E}_{n,n} & \ldots & E_n & \mathcal{E}_{n,n} & \mathcal{E}_{n,n} \\ \mathcal{E}_{n,n} & \ldots & \mathcal{E}_{n,n} & E_n & \mathcal{E}_{n,n} \end{bmatrix} \in \mathbb{R}^{nl \times nl}$$

for $k \in \mathbb{N}$.

Transportation systems usually operate according to a timetable demanding that a public vehicle is not allowed to depart before its scheduled departure time. A timetable can be incorporated in the MPL system (3) by adding an inhomogeneous term:

$$\tilde{\mathbf{x}}(k+1) = \tilde{A} \otimes \tilde{\mathbf{x}}(k) \oplus \tilde{\mathbf{d}}(k+1),$$

where $\tilde{\mathbf{d}}(k+1) = [\mathbf{d}(k+1)^T \quad \ldots \quad \mathbf{d}(k+2-l)^T]^T \in \mathbb{R}^{nl}$ is a vector containing the scheduled $(k+1)$th, kth, …, $(k+2-l)$th departure times. The dynamics of $\mathbf{d}(k)$ are assumed to be periodic (or regular), i. e. there exists a positive number $T \in \mathbb{R}$ such that $\mathbf{d}(k) = T^{\otimes k} \otimes \mathbf{d}(0)$. Let us restate the objective of this paper by using the previous notions. Our objective is determining period T and time of initial departure $\mathbf{d}(0)$ for any strongly connected public transportation network.

2.3. Graph representation of max-plus matrices

In this subsection, we briefly introduce graph representation of a max-plus matrix and some related notions [15].

A directed graph $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{D})$, where $\mathcal{V}$ is the set of vertices (or nodes) and $\mathcal{D} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Notation $(j, i) \in \mathcal{D}$ denotes an edge from node j to node i. Node j is called the origin of edge (j, i), whereas node i is called the destination of edge (j, i). Given a matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$, a directed graph $\mathcal{G}(A)$ is defined as a graph where the set of vertices $\mathcal{V} \overset{\text{def}}{=} \{1, 2, \ldots, n\}$ and the set of edges $\mathcal{D} \overset{\text{def}}{=} \{(j, i) : [A]_{i,j} \neq \varepsilon\}$. The entry $[A]_{i,j}$ represents the weight of edge $(j, i) \in \mathcal{D}$, for $i, j \in \underline{n}$. In the literature, this graph is called precedence or communication graph.

A path p from node i to node j in a graph is a sequence of nodes $p = (i_1, i_2, \ldots, i_{s+1})$ with $i_1 = i$ and $i_{s+1} = j$ such that $(i_k, i_{k+1}) \in \mathcal{D}$, for each $k \in \underline{s}$. This path has length s, which is denoted by $||p||_l = s$. The set of paths from i to j of length k is denoted by $P(i, j, k)$. A vertex j is said to be reachable from a vertex i, denoted by $i\mathcal{R}j$, if there exists a path from i to j. A strongly connected graph is a graph such that every vertex is reachable from any vertex. A matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$ is irreducible if $\mathcal{G}(A)$ is strongly connected.

A circuit of length s is a closed path, namely a path $p = (i_1, i_2, \ldots, i_{s+1})$ such that $i_1 = i_{s+1}$. A loop is a circuit consisting of one edge. An elementary circuit is a circuit in which $i_1, i_2, \ldots, i_s$ are distinct. The following proposition describes the relation between the power of a square matrix A and the existence of circuits in $\mathcal{G}(A)$.

Proposition 2.1. (Heidergott et al. [15]) Given $A \in \mathbb{R}_\varepsilon^{n\times n}$, the graph $\mathcal{G}(A)$ does not have any circuit if and only if $A^{\otimes k} = \mathcal{E}_{n,n}$ for all $k \geq n$.

2.4. Spectral and generalized eigenvalue problems

In this subsection, we introduce spectral problems, cycle-time vectors, periodic regimes and generalized eigenvalue problems over the max-plus algebra. The max-plus spectral problem is a problem of determining the max-plus eigenvalue and corresponding max-plus eigenvectors of a given square matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$ [2, 15]. This problem is related to the explicit first-order autonomous MPL systems (3) and can be solved by using the power algorithm [21].

Definition 2.2. (Eigenvalue and eigenvectors Baccelli et al. [2], Heidergott et al. [15]) Let $A \in \mathbb{R}_\varepsilon^{n\times n}$ be a max-plus matrix. Scalar $\lambda \in \mathbb{R}_\varepsilon$ and vector $\mathbf{v} \in \mathbb{R}_\varepsilon^n$ that contains at least one finite element are a max-plus eigenvalue and a corresponding max-plus eigenvector of A if $A \otimes \mathbf{v} = \lambda \otimes \mathbf{v}$ holds.

The power algorithm developed in [21] can be used to determine the max-plus eigenvalue and a corresponding max-plus eigenvector of matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$. The algorithm leverages recurrence relation $\mathbf{x}(k+1) = A \otimes \mathbf{x}(k)$ and uses finite initial condition $\mathbf{x}(0) \in \mathbb{R}^n$. Given matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$, the following algorithm computes the max-plus eigenvalue λ and a corresponding max-plus eigenvector $\mathbf{v}$:

1. Choose an arbitrary initial vector $\mathbf{x}(0) \in \mathbb{R}^n$;

2. Iterate $\mathbf{x}(k+1) = A \otimes \mathbf{x}(k)$ until there are integers $p > q \geq 0$ and a real number c such that a periodic behavior occurs, i. e. $\mathbf{x}(p) = c \otimes \mathbf{x}(q)$;

3. Compute the eigenvalue $\lambda = \dfrac{c}{p-q}$;

4. Compute one of the corresponding eigenvectors

$$\mathbf{v} = \bigoplus_{i=1}^{p-q} \left(\lambda^{\otimes(p-q-i)} \otimes \mathbf{x}(q+i-1) \right).$$

Let us determine the computational complexity of the power algorithm. We count the number of usual addition, usual multiplication, maximization and comparison operations in steps 2–4.

Step 2 This step consists of usual addition, maximization and comparison operations. In max-plus multiplication $A \otimes \mathbf{x}$, where $A \in \mathbb{R}_\varepsilon^{n\times n}$ and $\mathbf{x} \in \mathbb{R}_\varepsilon^n$, the number of usual addition and maximization operations is n^2 and $(n-1)n$, respectively. Since we compute $\mathbf{x}(1)$ until $\mathbf{x}(p)$, the total number of usual addition and maximization operations in this step is n^2p and $(n-1)np$, respectively. The comparison operation is used to find the value of q. When we compare two vectors of size n, there are n comparison operations. In the algorithm: $\mathbf{x}(1)$ is compared with $\mathbf{x}(0)$, $\mathbf{x}(2)$ is compared with $\mathbf{x}(1)$ and $\mathbf{x}(0)$, $\ldots$, $\mathbf{x}(p)$ is compared with $\mathbf{x}(0)$ until $\mathbf{x}(p-1)$. Thus the number of comparison operations associated with $\mathbf{x}(1)$, $\mathbf{x}(2)$, $\ldots$, $\mathbf{x}(p)$ is n, $2n$, $\ldots$, pn, respectively. The total number of comparison operations in this step is $p(p+1)n/2$.

Step 3 In this step, there are one usual addition and one usual multiplication operations.

Step 4 This step consists of usual addition, usual multiplication and maximization operations. For a particular value of i, there are $n+4$ usual addition operations: 2 operations in $p-q-i$, 2 operations in $q+i-1$, n operations in max-plus multiplication of a scalar and an n-dimensional vector. Thus for each i, the number of usual addition operations is $n+4$. It follows that the total number of usual addition operations in this step is $(n+4)(p-q)$. The usual multiplication is used to compute the max-plus power of λ, which is done once at each value of i. Thus the total number of usual multiplication operations in this step is $p-q$. The maximization operation is used to compute the max-plus addition of $p-q$ vectors of size n. Notice that there are $p-q-1$ maximization operations in computing max-plus addition of $p-q$ scalars. Thus the total number of maximization operations is $(p-q-1)n$.

For the power algorithm, the number of usual addition operations is $pn^2+1+(n+4)(p-q)$, the total number of usual multiplication operations is $1+(p-q)$, the total number of maximization operations is $(n-1)np+(p-q-1)n$, the total number of comparison operations is $(p+1)pn/2$.

Notice that the max-plus eigenvalue and corresponding max-plus eigenvectors exist if state matrix A is irreducible. If the state matrix is reducible, we can determine the cycle-time vector, under the condition that the state matrix is regular. A max-plus matrix is called regular if there exists at least a finite element in each row. Cycle-time vector $\boldsymbol{\eta} \in \mathbb{R}^n$ is defined as

$$\boldsymbol{\eta} = \lim_{k\to\infty} \frac{\mathbf{x}(k)}{k},$$

where $\mathbf{x}(0) \in \mathbb{R}^n$. Notice that the cycle-time vector does not depend on the initial state $\mathbf{x}(0)$. The cycle-time vector is associated with the periodic regime. The periodic regime is defined as the set of states such that the dynamics are periodic with period $\boldsymbol{\eta}$, i. e.

$$\{\mathbf{x} \mid \mathbf{x}(k+1) = \boldsymbol{\eta} + \mathbf{x}(k)\}.$$

If all entries of $\boldsymbol{\eta}$ are the same, we can use the power algorithm to determine the entry of cycle-time vector and one of the states in the periodic regime. This can be done by replacing the terms "max-plus eigenvalue" and "the corresponding max-plus eigenvector" by "entry of cycle-time vector" and "a state in the periodic regime," respectively. The interested reader is referred to [12] for the details.

Generalized max-plus eigenvalue problem is an extension of max-plus spectral problem in the sense that it is related to the implicit higher-order autonomous MPL systems (1). More precisely, generalized max-plus eigenvalue problem is a problem of determining max-plus eigenvalue of a given collection of finitely many square matrices $\{A_0, A_1, A_2, \ldots, A_l\} \subseteq \mathbb{R}_\varepsilon^{n\times n}$ [6, 19]. Let us discuss the problem in more detail. First we define $\mathbb{R}_\varepsilon[\gamma]$ as the collection of max-plus polynomials over γ, where the coefficients belong to $\mathbb{R}_\varepsilon$. Then we define the mapping $\mathcal{A} : \mathbb{R} \to \mathbb{R}_\varepsilon^{n\times n}$ as $\mathcal{A}(\gamma) = \bigoplus_{s=0}^{l} A_s \otimes \gamma^{\otimes s}$. Notice that the mapping is parameterized by $A_0, \ldots, A_l$ and can be equivalently represented as a max-plus polynomial matrix with entries in set $\mathbb{R}_\varepsilon[\gamma]$, i. e. $\mathcal{A} \in (\mathbb{R}_\varepsilon[\gamma])^{n\times n}$.

Definition 2.3. (Generalized eigenvalue problem Baccelli et al. [2], Cochet-Terrasson et al. [6], Soto y Koelemeijer [19]) Given a max-plus polynomial matrix $\mathcal{A} \in (\mathbb{R}_\varepsilon[\gamma])^{n\times n}$, determine $\lambda(\mathcal{A}) \in \mathbb{R}$ and vector $\mathbf{v} \in \mathbb{R}^n$ such that $\mathcal{A}(\lambda(\mathcal{A})^{\otimes -1}) \otimes \mathbf{v} = \mathbf{v}$. Alternatively, given $A_s \in \mathbb{R}_\varepsilon^{n\times n}$, for $s \in \{0, 1, \ldots, l\}$, determine $\lambda(\mathcal{A}) \in \mathbb{R}$ and vector $\mathbf{v} \in \mathbb{R}^n$ such that $\bigoplus_{s=0}^{l} A_s \otimes \lambda(\mathcal{A})^{\otimes -s} \otimes \mathbf{v} = \mathbf{v}$.

A max-plus polynomial matrix $\mathcal{A}(\gamma)$ is irreducible if max-plus matrix $\mathcal{A}(0)$ is irreducible, i. e. $\mathcal{G}(\mathcal{A}(0))$ is a strongly connected graph [2, Th. 3.28]. The following proposition states the sufficient conditions for the existence of generalized max-plus eigenvalues.

Proposition 2.4. (Baccelli et al. [2], Cochet-Terrasson et al. [6], Soto y Koelemeijer [19]) If $\mathcal{A}(\gamma) = \bigoplus_{s=0}^{l} A_s \otimes \gamma^{\otimes s}$ is an irreducible max-plus polynomial matrix and $\mathcal{G}(A_0)$ does not have any circuit, then $\mathcal{A}$ has a unique generalized max-plus eigenvalue, which is equal to the maximum cycle mean of $\mathcal{G}(\mathcal{A}(0))$.

2.5. Synchronization rules

Before constructing a schedule, first we have to define some synchronization rules. The aim of using synchronization rules is to ensure that passengers can travel from any position to any destination in the transportation system. Synchronization rule is explained in the following definition.

Definition 2.5. (Synchronization rules Fahim et al. [11]) All vehicles arriving at any station (public transportation stop) have to wait for each other to allow passengers to change vehicles.

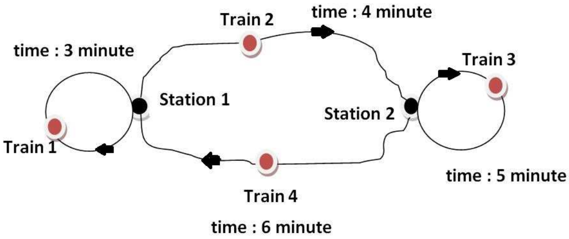

Fig. 1. A simple train network.

Let us consider the train network depicted in Figure 1. There are two stations (station 1 and station 2) and four trains (train 1, train 2, train 3 and train 4). The time needed by train 1 to go from station 1 to station 1 is 3 minutes. The complete traveling time can be seen in Figure 1. According to Definition 2.5, the synchronization rules are as follows:

- $(k+1)$th departure from station 1 to station 1 has to wait kth arrival train from station 1 to station 1 and from station 2 to station 1.
- $(k+1)$th departure from station 1 to station 2 has to wait kth arrival train from station 1 to station 1 and from station 2 to station 1.
- $(k+1)$th departure from station 2 to station 2 has to wait kth arrival train from station 2 to station 2 and from station 1 to station 2.
- $(k+1)$th departure from station 2 to station 1 has to wait kth arrival train from station 2 to station 2 and from station 1 to station 2.

According to above rules and assuming that all trains depart as soon as possible, the dynamics of departure time of all trains can be written as an explicit first-order autonomous MPL system (3):

$$\begin{aligned}
x_1(k+1) &= \max\{x_1(k)+3, x_4(k)+5\} = x_1(k)\otimes 3 \oplus x_4(k)\otimes 5\\
x_2(k+1) &= \max\{x_1(k)+3, x_4(k)+5\} = x_1(k)\otimes 3 \oplus x_4(k)\otimes 5\\
x_3(k+1) &= \max\{x_2(k)+4, x_3(k)+6\} = x_2(k)\otimes 4 \oplus x_3(k)\otimes 6\\
x_4(k+1) &= \max\{x_2(k)+4, x_3(k)+6\} = x_2(k)\otimes 4 \oplus x_3(k)\otimes 6
\end{aligned}$$

where $x_1(k+1)$, $x_2(k+1)$, $x_3(k+1)$, $x_4(k+1)$ are the time of $(k+1)$th departure from station 1 to station 1, from station 1 to station 2, from station 2 to station 2 and from station 2 to station 1, respectively. As it will be clear in Section 3.3, the MPL system will be used to determine a regular schedule of train departure.

3. MODELING AND SCHEDULING OF PUBLIC TRANSPORTATION NETWORKS

In this section, first we construct a model of public transportation networks as MPL systems and analyze its properties (cf. Section 3.1). Then we reduce size of the model and analyze the impact of reduction w.r.t. the max-plus eigenvectors in Section 3.2. In Section 3.3, we describe a procedure to construct a regular schedule. Finally, we illustrate the procedure on a case study (cf. Section 3.4).

3.1. Modeling of public transportation networks

Initially we define the formal representation of a public transportation network. Then we construct a model of public transportation networks as an implicit higher-order MPL system (1). Next we show that the corresponding max-plus polynomial matrix is irreducible. It follows that the max-plus polynomial matrix has a unique generalized max-plus eigenvalue. Finally we transform the model as an explicit first-order MPL system (3) by using a novel technique.

Definition 3.1. (Road network of public transportations) The road network of public transportations is represented as max-plus matrix $N \in \mathbb{R}_{\varepsilon}^{m\times m}$, where m denotes the number of stations. The entries of N represent the traveling time. More precisely if $[N]_{i,j} \neq \varepsilon$, it denotes the traveling time from station j to i, otherwise if $[N]_{i,j} = \varepsilon$, there is no direct route from station j to i. We define $\mathcal{V} = \{1, 2, \ldots, m\}$ and $\mathcal{D} = \{(i, j) \mid [N]_{j,i} \neq \varepsilon\}$ as the set of vertices and edges of $\mathcal{G}(N)$, respectively. Let $\{e_1, e_2, \ldots, e_n\}$ represents the set of routes. The origin and destination of route e_u are denoted by i_u and j_u respectively, for $u \in \underline{n}$. In other words, $e_u = (i_u, j_u)$ for $u \in \underline{n}$. Finally let p_u be the number of public vehicles serving route e_u initially, for $u \in \underline{n}$.

In the following lemma, we construct an MPL system from a road network of public transportations. The event is defined as the vehicle departure on a route. Since the number of routes in the network is n, we obtain an n-dimensional MPL system.

Lemma 3.2. Let $N \in \mathbb{R}_{\varepsilon}^{m\times m}$ denotes a road network of public transportations where $\mathcal{G}(N)$ is strongly connected. The time of vehicle departures in all routes can be modeled as the following implicit higher-order MPL system

$$x_u(k+1) = \bigoplus_{\substack{q=1\\ j_q=i_u}}^{n} [N]_{j_q,i_q} \otimes x_q(k+1-p_u), \qquad u \in \underline{n} \tag{4}$$

where $x_u(k+1)$ is the time of $(k+1)$th departure on route e_u, i. e. from station (vertex) i_u to station (vertex) j_u, for $u \in \underline{n}$. In matrix notation, (4) can be rewritten as

$$\mathbf{x}(k+1) = \bigoplus_{s=0}^{p_{\max}} A_s \otimes \mathbf{x}(k+1-s) \tag{5}$$

where $\mathbf{x}(k) = [x_1(k) \quad \ldots \quad x_n(k)]^T \in \mathbb{R}^n$, $p_{\max} = \max\{p_1, p_2, \ldots, p_n\}$ and

$$[A_s]_{a,b} = \begin{cases} [N]_{i_a,i_b} & , \text{ if } p_a = s, \\ \varepsilon & , \text{ otherwise.} \end{cases} \tag{6}$$

Proof. We show that for any route, we obtain the model in (4). More precisely, we consider three route categories depending on the initial number of public vehicles serving the route: $p_u = 0$, $p_u = 1$ and $p_u \geq 2$.

($p_u = 1$) In the first case, we assume that the initial number of public vehicles serving route e_u is 1. This case is illustrated in Figure 2. In this figure, the routes $e_{u_1}, e_{u_2}, \ldots, e_{u_r}$ are edges in $\mathcal{D}$ such that $j_{u_1} = j_{u_2} = \cdots = j_{u_r} = i_u$ and b_1 is the vehicle serving route e_u initially.

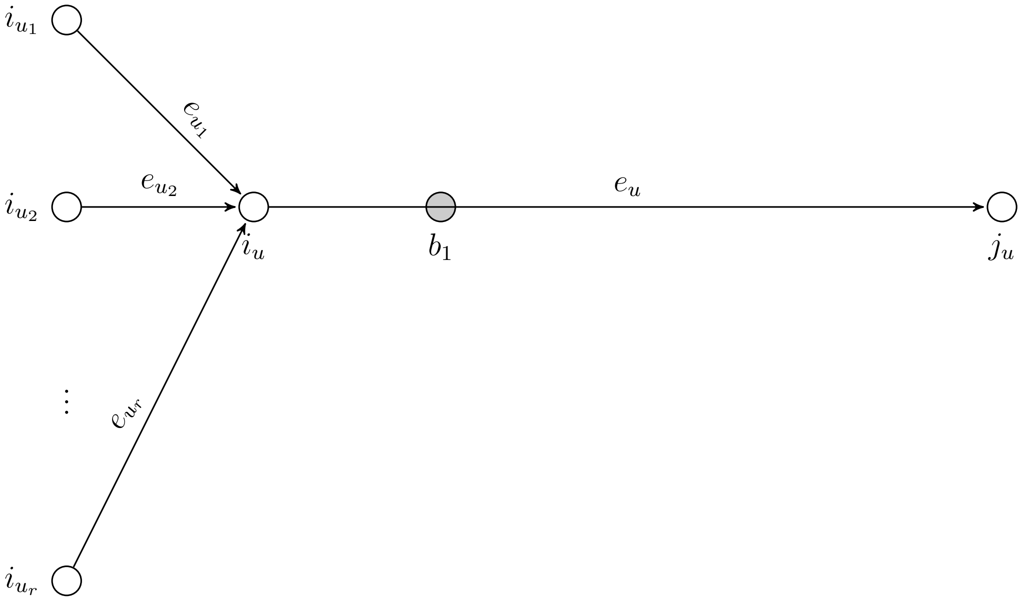

Fig. 2. The routes that go to vertex i_u are e_{u_1}, e_{u_2}, $\ldots$, e_{u_r}. Initially, the number of public vehicles serving route e_u is 1.

By using Definition 2.5, the departure schedule of route e_u depends on other routes which go to vertex i_u. More precisely, the $(k+1)$th departure from station i_u to station j_u has to wait the kth arrival from stations i_{u_1}, i_{u_2}, $\ldots$, i_{u_r}. Thus we obtain

$$x_u(k+1) = [N]_{i_u,i_{u_1}} \otimes x_{u_1}(k) \oplus [N]_{i_u,i_{u_2}} \otimes x_{u_2}(k) \oplus \cdots \oplus [N]_{i_u,i_{u_r}} \otimes x_{u_r}(k)$$

which can be rewritten as

$$x_u(k+1) = \bigoplus_{\substack{q=1 \\ j_q = i_u}}^{n} [N]_{j_q,i_q} \otimes x_q(k+1-p_u)$$

where $p_u = 1$.

($p_u \geq 2$) In the second case, we assume that the initial number of public vehicles serving route e_u is t, where $t \geq 2$. This case is illustrated in Figure 3. In this figure, the

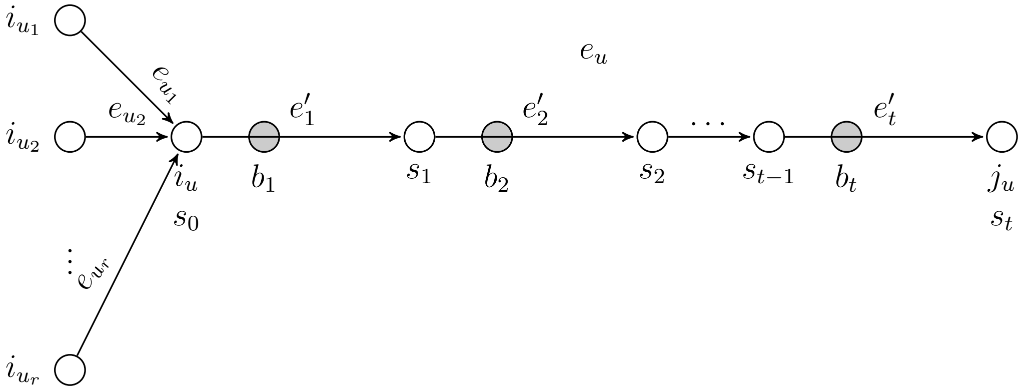

Fig. 3. The routes that go to i_u are e_{u_1}, e_{u_2}, ..., e_{u_r}. Initially, the number of public vehicles serving route e_u is t.

routes e_{u_1}, e_{u_2}, ..., e_{u_r} are edges in $\mathcal{D}$ such that $j_{u_1} = j_{u_2} = \cdots = j_{u_r} = i_u$ and $b_1, b_2, \ldots, b_t$ are vehicles serving route e_u initially.

Our approach to construct the model consists of two steps. In the first step, we construct a first-order model by defining new stations and routes. More precisely, we define $t-1$ virtual stations s_1, s_2, ..., s_{t-1} which are located between i_u and j_u (cf. Figure 3). Furthermore we define t routes e'_1, e'_2, ..., e'_t where e'_i is the route from s_{i-1} to s_i for $i \in \underline{t}$. Notice that $i_u = s_0$ and $j_u = s_t$. The traveling time of route e'_i is 0 if $i \in \underline{t-1}$ and $[N]_{j_u,i_u}$ if $i = t$. Initially vehicle b_i is serving route e'_i for $i \in \underline{t}$. Then we introduce new departure variables x'_1, x'_2, ..., x'_t where $x'_i(k)$ is the time of the kth departure in route e'_i for $i \in \underline{t}$. By using Definition 2.5, we obtain the following recurrence relation

$$\begin{aligned} x'_1(k+1) &= [N]_{i_u,i_{u_1}} \otimes x_{u_1}(k) \oplus \cdots \oplus [N]_{i_u,i_{u_r}} \otimes x_{u_r}(k) \\ x'_2(k+1) &= x'_1(k) \\ x'_3(k+1) &= x'_2(k) \\ &\vdots \\ x'_t(k+1) &= x'_{t-1}(k). \end{aligned}$$

In the second step, we construct a higher-order model which describes the dynamics of departure time in route e'_t. More precisely, we construct an expression for x'_t that is a function of x_{u_1}, x_{u_2}, ..., x_{u_r}. To obtain such expression, we substitute the above equations to the last equation:

$$\begin{aligned} x'_t(k+1) = {} & [N]_{i_u,i_{u_1}} \otimes x_{u_1}(k+1-t) \oplus [N]_{i_u,i_{u_2}} \otimes x_{u_2}(k+1-t) \\ & \oplus \cdots \oplus [N]_{i_u,i_{u_r}} \otimes x_{u_r}(k+1-t). \end{aligned}$$

We define the departure time in route e_u as the departure time in route e'_t, i. e. $x_u(k+1) = x'_t(k+1)$. It follows that the previous equation can be rewrittens

$$x_u(k+1) = \bigoplus_{\substack{q=1 \\ j_q = i_u}}^{n} [N]_{j_q,i_q} \otimes x_q(k+1-p_u)$$

($p_u = 0$) In the third case, we assume that the initial number of public vehicles serving route e_u is 0. This case is illustrated in Figure 4. In this figure, the routes e_{u_1}, e_{u_2}, …, e_{u_r} are edges in $\mathcal{D}$ such that $j_{u_1} = j_{u_2} = \cdots = j_{u_r} = i_u$.

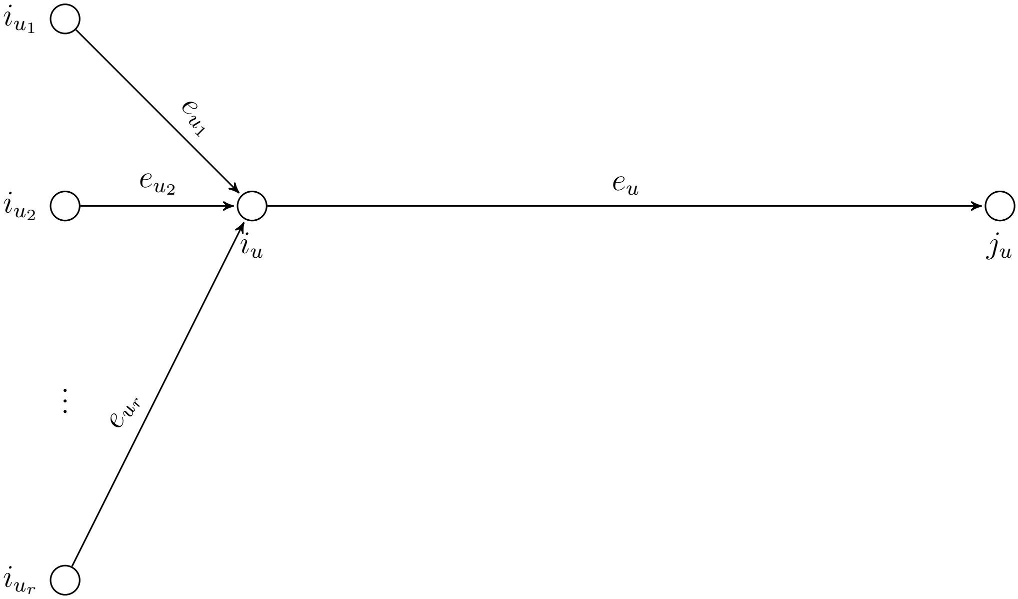

Fig. 4. The routes that go to i_u are e_{u_1}, e_{u_2}, …, e_{u_r}. Initially, the number of public vehicles serving route e_u is 0.

By using Definition 2.5, the departure schedule of route e_u depends on other routes which go to vertex i_u. More precisely, the $(k+1)$th departure from station i_u to station j_u has to wait the $(k+1)$th arrival from stations i_{u_1}, i_{u_2}, …, i_{u_r}. Thus we obtain

$$\begin{aligned} x_u(k+1) = [N]_{i_u,i_{u_1}} \otimes x_{u_1}(k+1) \oplus [N]_{i_u,i_{u_2}} \otimes x_{u_2}(k+1) \\ \oplus \cdots \oplus [N]_{i_u,i_{u_r}} \otimes x_{u_r}(k+1) \end{aligned}$$

which can be rewritten as

$$x_u(k+1) = \bigoplus_{\substack{q=1 \\ j_q=i_u}}^{n} [N]_{j_q,i_q} \otimes x_q(k+1-p_u)$$

where $p_u = 0$.

Notice that in all cases, we obtain a model defined in (4). □

Next we derive the uniqueness of generalized max-plus eigenvalue in the above model (5). Lemma 3.3 proves that $\mathcal{G}(A_0)$ does not have any circuit. Lemma 3.4 shows that max-plus polynomial matrix $\mathcal{A}(\gamma)$ is irreducible. Finally according to Proposition 2.4, the model (5) has a unique generalized max-plus eigenvalue.

Lemma 3.3. Graph $\mathcal{G}(A_0)$ does not have any circuit where A_0 is defined in (5).

Proof. We prove the lemma by contradiction. We assume that graph $\mathcal{G}(A_0)$ has a circuit. Suppose that the circuit consists of the following edges $e_{u_1}, e_{u_2}, \ldots, e_{u_r}$ where $j_{u_1} = i_{u_2}$, $j_{u_2} = i_{u_3}$, $\ldots$, $j_{u_{r-1}} = i_{u_r}$ and $j_{u_r} = i_{u_1}$. Notice that $p_{u_1} = p_{u_2} = \cdots = p_{u_r} = 0$. From (4), we know that

$$\begin{aligned} x_{u_1}(k+1) &= \left(\bigoplus_{\substack{q=1 \\ j_q=i_{u_1}, i_q \neq i_{u_r}}}^{n} [N]_{j_q,i_q} \otimes x_q(k+1) \right) \oplus [N]_{j_{u_r},i_{u_r}} \otimes x_{u_r}(k+1) \\ &= \Re_1 \oplus [N]_{j_{u_r},i_{u_r}} \otimes x_{u_r}(k+1) \qquad (7) \\ x_{u_2}(k+1) &= \left(\bigoplus_{\substack{q=1 \\ j_q=i_{u_2}, i_q \neq i_{u_1}}}^{n} [N]_{j_q,i_q} \otimes x_q(k+1) \right) \oplus [N]_{j_{u_1},i_{u_1}} \otimes x_{u_1}(k+1) \\ &= \Re_2 \oplus [N]_{j_{u_1},i_{u_1}} \otimes x_{u_1}(k+1) \qquad (8) \\ x_{u_3}(k+1) &= \left(\bigoplus_{\substack{q=1 \\ j_q=i_{u_3}, i_q \neq i_{u_2}}}^{n} [N]_{j_q,i_q} \otimes x_q(k+1) \right) \oplus [N]_{j_{u_2},i_{u_2}} \otimes x_{u_2}(k+1) \\ &= \Re_3 \oplus [N]_{j_{u_2},i_{u_2}} \otimes x_{u_2}(k+1) \qquad (9) \\ &\vdots \\ x_{u_r}(k+1) &= \left(\bigoplus_{\substack{q=1 \\ j_q=i_{u_r}, i_q \neq i_{u_{r-1}}}}^{n} [N]_{j_q,i_q} \otimes x_q(k+1) \right) \oplus [N]_{j_{u_{r-1}},i_{u_{r-1}}} \otimes x_{u_{r-1}}(k+1) \\ &= \Re_r \oplus [N]_{j_{u_{r-1}},i_{u_{r-1}}} \otimes x_{u_{r-1}}(k+1). \qquad (10) \end{aligned}$$

First we substitute (7) to (8), which yields

$$x_{u_2}(k+1) = \Re_2 \oplus [N]_{j_{u_1},i_{u_1}} \otimes \Re_1 \oplus [N]_{j_{u_1},i_{u_1}} \otimes [N]_{j_{u_r},i_{u_r}} \otimes x_{u_r}(k+1). \tag{11}$$

Then we substitute (11) to (9), which produces

$$\begin{aligned} x_{u_3}(k+1) = {} & \Re_3 \oplus [N]_{j_{u_2},i_{u_2}} \otimes \Re_2 \oplus [N]_{j_{u_2},i_{u_2}} \otimes [N]_{j_{u_1},i_{u_1}} \otimes \Re_1 \\ & \oplus [N]_{j_{u_2},i_{u_2}} \otimes [N]_{j_{u_1},i_{u_1}} \otimes [N]_{j_{u_r},i_{u_r}} \otimes x_{u_r}(k+1). \end{aligned}$$

If we continue the substitution process until (10), we obtain

$$\begin{aligned} x_{u_r}(k+1) &= \left(\bigoplus_{k=0}^{r-1} \Re_{r-k} \otimes \bigotimes_{s=1}^{k} [N]_{j_{u_{r-s}},i_{u_{r-s}}} \right) \oplus x_{u_r}(k+1) \bigotimes_{s=1}^{r} [N]_{j_{u_s},i_{u_s}} \\ &= \Re \oplus [N]_{j_{u_1},i_{u_1}} \otimes [N]_{j_{u_2},i_{u_2}} \otimes \cdots \otimes [N]_{j_{u_r},i_{u_r}} \otimes x_{u_r}(k+1). \end{aligned}$$

From the above equation, we can infer that

$$[N]_{j_{u_1},i_{u_1}} \otimes [N]_{j_{u_2},i_{u_2}} \otimes \cdots \otimes [N]_{j_{u_r},i_{u_r}} \otimes x_{u_r}(k+1) \leq x_{u_r}(k+1),$$

which is equivalent with $[N]_{j_{u_1},i_{u_1}} \otimes [N]_{j_{u_2},i_{u_2}} \otimes \cdots \otimes [N]_{j_{u_r},i_{u_r}} \leq 0$. This contradicts the fact that the traveling time is a positive number. □

Lemma 3.4. Max-plus polynomial matrix $\mathcal{A}(\gamma) = \bigoplus_{s=0}^{p_{\max}} A_s \otimes \gamma^{\otimes s}$ is irreducible where A_0, A_1, …, $A_{p_{\max}}$ are defined in (5).

Proof. In this proof, we define $\mathcal{V}(\mathcal{A}(0)) = \{1, 2, \ldots, n\}$ as the set of vertices of graph $\mathcal{G}(\mathcal{A}(0))$. Notice that the set of vertices of graph $\mathcal{G}(\mathcal{A}(0))$ corresponds to the set of edges of graph $\mathcal{G}(N)$. Let $\mathcal{V}(N) = \{i_1, j_1, i_2, j_2, \ldots, i_n, j_n\}$ be the set of vertices of graph $\mathcal{G}(N)$.

Remember that $\mathcal{A}(\gamma)$ is irreducible if $\mathcal{G}(\mathcal{A}(0))$ is strongly connected (cf. Section 2.4). We will prove that $\mathcal{G}(\mathcal{A}(0))$ is strongly connected. Let l, m be arbitrary vertices in $\mathcal{G}(\mathcal{A}(0))$. We will show that there exists a path from l to m in $\mathcal{G}(\mathcal{A}(0))$. Since $\mathcal{G}(N)$ is strongly connected and i_l, i_m are vertices in $\mathcal{G}(N)$, there exists a path from i_l to i_m in $\mathcal{G}(N)$. Without loss of generality, we assume the path consists of the following edges (i_l, i_{r_1}), (i_{r_1}, i_{r_2}), (i_{r_2}, i_{r_3}), …, $(i_{r_{t-1}}, i_{r_t})$, (i_{r_t}, i_m). This means $[N]_{i_{r_1},i_l}$, $[N]_{i_{r_2},i_{r_1}}$, $[N]_{i_{r_3},i_{r_2}}$, …, $[N]_{i_{r_t},i_{r_{t-1}}}$, $[N]_{i_m,i_{r_t}}$ are finite. Since $\mathcal{A}(0) = A_0 \oplus A_1 \oplus \ldots A_{p_{\max}}$, according to (6), we know that $[\mathcal{A}(0)]_{a,b} = [N]_{i_a,i_b}$ for $a, b \in \underline{n}$. It follows that $[\mathcal{A}(0)]_{r_1,l}$, $[\mathcal{A}(0)]_{r_2,r_1}$, $[\mathcal{A}(0)]_{r_3,r_2}$, …, $[\mathcal{A}(0)]_{r_t,r_{t-1}}$, $[\mathcal{A}(0)]_{m,r_t}$ are finite. In other words, there exists a path $(l, r_1, r_2, r_3, \ldots, r_{t-1}, r_t, m)$ from l to m in $\mathcal{G}(\mathcal{A}(0))$. □

In order to synthesize a regular schedule, the MPL system has to be explicit and first order. Thus in the remainder of this section, we will construct an explicit first-order MPL system. The approach consists of two steps. In the first step, we transform the explicit MPL system to an implicit MPL system (cf. Section 2.2). In the second step, we use a novel approach to transform the higher-order MPL system to a first-order MPL system. The motivation of using the novel approach is that we are able to construct a systematic procedure for the reduction process (cf. Section 3.2). To the best of our knowledge, it is not possible to construct such procedure using the well-known approach (cf. Section 2.2).

Let us focus on the first step. Proposition 2.1 and Lemma 3.3 imply A_0^* exists. Thus the implicit higher-order MPL system (5) can be transformed as the following explicit higher-order MPL system

$$\mathbf{x}(k+1) = \bigoplus_{s=1}^{p_{\max}} A_0^* \otimes A_s \otimes \mathbf{x}(k+1-s). \tag{12}$$

With regards to the second step, the explicit first-order MPL system is constructed by defining augmented state $\tilde{\mathbf{z}}(k) = [\mathbf{z}_1(k)^T \quad \dots \quad \mathbf{z}_{p_{\max}}(k)^T]^T \in \mathbb{R}^{n(p_{\max})}$. The state equation is defined as

$$\tilde{\mathbf{z}}(k+1) = \tilde{W} \otimes \tilde{\mathbf{z}}(k), \tag{13}$$

where the state matrix is given by

$$\tilde{W} = \begin{bmatrix} A_0^* \otimes A_1 & E_n & \mathcal{E}_{n,n} & \dots & \mathcal{E}_{n,n} \\ A_0^* \otimes A_2 & \mathcal{E}_{n,n} & E_n & \dots & \mathcal{E}_{n,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_0^* \otimes A_{p_{\max}-1} & \mathcal{E}_{n,n} & \mathcal{E}_{n,n} & \dots & E_n \\ A_0^* \otimes A_{p_{\max}} & \mathcal{E}_{n,n} & \mathcal{E}_{n,n} & \dots & \mathcal{E}_{n,n} \end{bmatrix}$$

for $k \in \mathbb{N}$. As a side note, notice that $\tilde{W} \in \mathbb{R}^{n(p_{\max}) \times n(p_{\max})}$.

Proposition 3.5. The dynamics of $\mathbf{z}_1(k+1)$ in (13) coincide with the dynamics of $\mathbf{x}(k+1)$ in (12) for $k+1 \geq p_{\max}$.

Proof. From the first n equations in (13), we know that

$$\mathbf{z}_1(k+1) = A_0^* \otimes A_1 \otimes \mathbf{z}_1(k) \oplus \mathbf{z}_2(k).$$

Then we substitute the $(n+1)$th equation until $(2n)$th equation in (13) to the term $\mathbf{z}_2(k)$ in the above equation. The result is

$$\mathbf{z}_1(k+1) = A_0^* \otimes A_1 \otimes \mathbf{z}_1(k) \oplus A_0^* \otimes A_2 \otimes \mathbf{z}_1(k-1) \oplus \mathbf{z}_3(k-1).$$

If we continue the substitution until the last n equations, we obtain

$$\mathbf{z}_1(k+1) = A_0^* \otimes A_1 \otimes \mathbf{z}_1(k) \oplus A_0^* \otimes A_2 \otimes \mathbf{z}_1(k-1) \oplus \dots \oplus A_0^* \otimes A_{p_{\max}} \otimes \mathbf{z}_1(k+1-p_{\max}),$$

which is the same with (12). Notice that the equation above is well defined since $k+1 \geq p_{\max}$, or equivalently $k+1-p_{\max} \geq 0$. □

Proposition 3.5 implies the max-plus eigenvalue and max-plus eigenvectors of $\mathbf{z}_1(k+1)$ in (13) coincide with the max-plus eigenvalue and max-plus eigenvectors in (12). Since the model in (12) has a unique max-plus eigenvalue, $\mathbf{z}_1(k+1)$ in (13) also has a unique max-plus eigenvalue. Let us note that the dynamics of $\mathbf{z}(k+1)$ for $k+1 < p_{\max}$ do not affect the max-plus eigenvalue and eigenvectors because both max-plus eigenvalue and max-plus eigenvectors are associated with the steady-state behavior. Notice that the steady-state behavior is related with behavior of the state when $k \to \infty$.

By using substitution similar to the proof of Proposition 3.5, one can show that the dynamics for $\mathbf{z}_1, \ldots, \mathbf{z}_{p_{\max}}$ are given by

$$\mathbf{z}_i(k+1) = \bigoplus_{j=i}^{p_{\max}} A_0^* \otimes A_j \otimes \mathbf{z}_1(k+2-j), \qquad \text{for all } i \in \underline{p_{\max}}. \tag{14}$$

3.2. Reduced model of public transportation networks

In this section, we describe two reduction procedures. Initially we describe the first procedure to reduce size of the state matrix of an MPL system. Then we determine size of the state matrix in the MPL system that models a public transportation network under some condition. Finally we discuss the second reduction procedure.

Lemma 3.6. Let $H \in \mathbb{R}_\varepsilon^{n\times n}$ where all entries in the tth row are ε, i. e. $[H]_{t,j} = \varepsilon$ for all $j \in \underline{n}$. We define H' as the matrix obtained by removing the tth row and the tth column of matrix H. The following statements hold:

1. If $\lambda \in \mathbb{R}$ and $\mathbf{v} = [v_1 \quad v_2 \quad \ldots \quad v_n]^T \in \mathbb{R}_\varepsilon^n$ are a max-plus eigenvalue and a corresponding max-plus eigenvector of matrix H respectively, then the tth entry of $\mathbf{v}$ is ε, i. e. $v_t = \varepsilon$.

2. Scalar $\lambda \in \mathbb{R}$ and vector $\mathbf{v} = [v_1 \quad v_2 \quad \ldots \quad v_n]^T \in \mathbb{R}_\varepsilon^n$ are a max-plus eigenvalue and a corresponding max-plus eigenvector of matrix H respectively if and only if λ and $\mathbf{v}' = [v_1 \quad \ldots \quad v_{t-1} \quad v_{t+1} \quad \ldots \quad v_n]^T \in \mathbb{R}_\varepsilon^{n-1}$ are a max-plus eigenvalue and a corresponding max-plus eigenvector of matrix H' respectively.

Proof.

1. Since λ and $\mathbf{v} = [v_1 \quad v_2 \quad \ldots \quad v_n]^T$ are a max-plus eigenvalue and a corresponding max-plus eigenvector of matrix H respectively, from Definition 2.2 we know that

$$\bigoplus_{j=1}^{n} [H]_{i,j} \otimes v_j = \lambda \otimes v_i, \qquad \text{for all } i \in \underline{n}.$$

For $i = t$, we obtain $\bigoplus_{j=1}^{n} [H]_{t,j} \otimes v_j = \lambda \otimes v_i$. Because $[H]_{t,j} = \varepsilon$ for all $j \in \underline{n}$, then the preceding equation becomes $\lambda \otimes v_t = \varepsilon$. Since $\lambda \in \mathbb{R}$, we obtain $v_t = \varepsilon$.

2. First we focus on the "if" part. Let λ and $\mathbf{v} = [v_1 \quad v_2 \quad \ldots \quad v_n]^T$ be a max-plus eigenvalue and a corresponding max-plus eigenvector of matrix H, respectively. From Definition 2.2, we obtain

$$\bigoplus_{j=1}^{n} [H]_{i,j} \otimes v_j = \lambda \otimes v_i, \qquad \text{for all } i \in \underline{n}. \tag{15}$$

Equation (15) can be written as

$$\left(\bigoplus_{j=1}^{t-1}[H]_{i,j}\otimes v_j\right)\oplus([H]_{i,t}\otimes v_t)\oplus\left(\bigoplus_{j=t+1}^{n}[H]_{i,j}\otimes v_j\right)=\lambda\otimes v_i. \tag{16}$$

From the proof of the first item, we know that $v_t=\varepsilon$. Thus (16) is equivalent with

$$\left(\bigoplus_{j=1}^{t-1}[H]_{i,j}\otimes v_j\right)\oplus\left(\bigoplus_{j=t+1}^{n}[H]_{i,j}\otimes v_j\right)=\lambda\otimes v_i, \tag{17}$$

for $i\in\{1,2,\dots,t-1,t+1,\dots,n\}$. By using matrix notation, (17) can be rewritten as

$$H'\otimes\mathbf{v}'=\lambda\otimes\mathbf{v}'.$$

Finally by reversing the direction, i. e. from bottom to up, we can prove the "only if" part.

□

Lemma 3.6 can be used to reduce the size of $\tilde{W}$ in (13). Let a route $u\in\underline{n}$ be arbitrary but fixed. Notice that the uth, $(u+n)$th, $\dots(u+(p_{\max}-1)n)$th rows are associated with the uth route. If we can find $t\in\{0,1,\dots,p_{\max}-1\}$ such that the entries in $(u+tn)$th, $(u+(t+1)n)$th, $\dots(u+(p_{\max}-1)n)$th rows for the first n columns are ε, then we can remove those rows, i. e. the $(u+tn)$th, $(u+(t+1)n)$th, $\dots(u+(p_{\max}-1)n)$th rows. Given state matrix $\tilde{W}$, the procedure to determine the rows that can be removed is as follows:

```
Require: $\tilde{W}$ the state matrix
Ensure: $R$ the set of row indices that can be removed
  $R := \emptyset$
  for $u := 1$ to $n$ do
    $t := p_{\max}-1$
    while $t \geq 0$ and $[\tilde{W}]_{u+tn,j}=\varepsilon$ for all $j\in\underline{n}$ do
      $R := R\cup\{u+tn\}$
      $t := t-1$
    end while
  end for
```

The above procedure works in all cases. Let us determine the worst-case computational complexity of the above procedure w.r.t. the comparison, usual addition and usual multiplication operations:

- First we focus on comparison operations. There are two comparison statements $t\geq 0$ and $[\tilde{W}]_{u+tn,j}=\varepsilon$. Comparison statement $t\geq 0$ is executed for each value of u and t. Thus, comparison statement $t\geq 0$ is executed $p_{\max}n$ times. Comparison statement $[\tilde{W}]_{u+tn,j}=\varepsilon$ is executed for each value of u, t and j. Thus, comparison statement $[\tilde{W}]_{u+tn,j}=\varepsilon$ is executed $p_{\max}n^2$ times. As a consequence, the total number of comparison operations is $p_{\max}n+p_{\max}n^2$.

- Then we continue to usual addition operations. There are three usual addition operations $p_{\max} - 1$, $u + tn$ and $t - 1$. Usual addition operation $p_{\max} - 1$ is executed once. Usual addition operations $u + tn$ and $t - 1$ are executed for each value of u and t. Thus, the number of usual addition operations is $2p_{\max}n$. Finally, the total number of usual addition operations is $2p_{\max}n + 1$.

- We focus on usual multiplication operations. There is one usual multiplication operation tn, which is executed $p_{\max}$ times. Thus, the total number of usual multiplication operations is $p_{\max}$.

As a special case, if $\min\{p_1, p_2, \dots, p_n\} = 1$, we can determine size of the reduced matrix, as shown in the following lemma.

Lemma 3.7. Matrix $\tilde{W}$ in (13) can be reduced to $\tilde{W}_r$ of size $(\sum_{u=1}^{n} p_u) \times (\sum_{u=1}^{n} p_u)$ if $\min\{p_1, p_2, \dots, p_n\} = 1$.

Proof. Since $\min\{p_1, p_2, \dots, p_n\} = 1$, then $A_0 = \mathcal{E}_{n,n}$. It follows that $A_0^* = E_n$. Thus matrix $\tilde{W}$ becomes

$$\tilde{W} = \begin{bmatrix} A_1 & E_n & \mathcal{E}_{n,n} & \dots & \mathcal{E}_{n,n} \\ A_2 & \mathcal{E}_{n,n} & E_n & \dots & \mathcal{E}_{n,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{p_{\max}-1} & \mathcal{E}_{n,n} & \mathcal{E}_{n,n} & \dots & E_n \\ A_{p_{\max}} & \mathcal{E}_{n,n} & \mathcal{E}_{n,n} & \dots & \mathcal{E}_{n,n} \end{bmatrix}$$

Let $u \in \underline{n}$ be an arbitrary but fixed route index. Remember that p_u is the initial number of vehicles in the uth route. From (6), the uth row of A_{p_u} has at least a finite entry. Equivalently, the $(u + (p_u - 1)n)$th row of $\tilde{W}$ in the first n columns has at least a finite entry. Furthermore for $t \in \{p_u, \dots, p_{\max} - 1\}$, the $(u + tn)$th row of $\tilde{W}$ does not have any finite element in the first n columns. Thus those rows can be removed. We conclude that the uth route needs p_u rows. It follows that the whole routes require $(\sum_{u=1}^{n} p_u)$ rows. □

The second approach to reduce size of the state matrix is based on the dynamics. If there are more than one states with the same dynamics, those states can be merged. Let us illustrate this by using the MPL system in Section 2.5:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 3 & \varepsilon & \varepsilon & 5 \\ 3 & \varepsilon & \varepsilon & 5 \\ \varepsilon & 4 & 6 & \varepsilon \\ \varepsilon & 4 & 6 & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix}$$

Notice that the dynamics of x_1 and x_2 are the same. The value of $x_1(0)$ and $x_2(0)$ can be different. However starting from $k = 1$, we can guarantee that $x_1(k) = x_2(k)$. Thus we can merge x_1 and x_2. By using a similar reasoning, we can merge x_3 and x_4. The interpretation is as follows. Notice that there are two trains that depart from each station. The previous observations mean that the departure time of both trains in any station is the same. Thus we can represent the departure time of both trains from a station as a single variable, rather than two variables. Since there are two stations, we need two state variables, called x'_1 and x'_2. We define $x'_1 = x_1 = x_2$ and $x'_2 = x_3 = x_4$. Thus the state equation for x_1 and x_3 in the original model can be written as follows:

$$\begin{aligned}
x'_1(k+1) &= 3 \otimes x'_1(k) \oplus \varepsilon \otimes x'_1(k) \oplus \varepsilon \otimes x'_2(k) \oplus 5 \otimes x'_2(k) \\
&= (3 \oplus \varepsilon) \otimes x'_1(k) \oplus (\varepsilon \oplus 5) \otimes x'_2(k) \\
&= 3 \otimes x'_1(k) \oplus 5 \otimes x'_2(k) \\
x'_2(k+1) &= \varepsilon \otimes x'_1(k) \oplus 4 \otimes x'_1(k) \oplus 6 \otimes x'_2(k) \oplus \varepsilon \otimes x'_2(k) \\
&= (\varepsilon \oplus 4) \otimes x'_1(k) \oplus (6 \oplus \varepsilon) \otimes x'_2(k) \\
&= 4 \otimes x'_1(k) \oplus 6 \otimes x'_2(k)
\end{aligned}$$

By using matrix notation, the reduced model can be re-written as

$$\begin{bmatrix} x'_1(k+1) \\ x'_2(k+1) \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 4 & 6 \end{bmatrix} \otimes \begin{bmatrix} x'_1(k) \\ x'_2(k) \end{bmatrix}$$

As mentioned previously, $x'_1(k+1)$ is the time of $(k+1)$th departure of both trains from station 1, whereas $x'_2(k+1)$ is the time of the $(k+1)$th departure of both trains from station 2. We do not write the general reduction procedure because one can generalize the reduction procedure from this simple example without any difficulty.

3.3. Scheduling of public transportation networks

This section discusses an algorithm to construct a periodic (or regular) schedule for public transportation networks. We cannot guarantee that the state matrix of the MPL system is irreducible. Thus we assume that the state matrix is reducible. As mentioned in Section 2.4, the algorithm requires that all elements of the cycle-time vector are the same if the state matrix is reducible. Lemma 3.8 shows that all elements of the cycle-time vector of the MPL system are the same.

Lemma 3.8. All elements of the cycle-time vector of the MPL system in (13) are the same.

Proof. Let $\tilde{\boldsymbol{\eta}} = [\boldsymbol{\eta}_1^T \ \ \dots \ \ \boldsymbol{\eta}_{p_{\max}}^T]^T \in \mathbb{R}^{n(p_{\max})}$ denotes the cycle-time vector and $\tilde{\mathbf{z}}(k) = [\mathbf{z}_1(k)^T \ \ \dots \ \ \mathbf{z}_{p_{\max}}(k)^T]^T \in \mathbb{R}^{n(p_{\max})}$ be a vector in the periodic regime for some value of k. Since $\mathbf{z}_1$ has a unique max-plus eigenvalue (cf. Section 3.1), all entries of $\boldsymbol{\eta}_1$ are the same. The entry of $\boldsymbol{\eta}_1$ is denoted by η_1. Since η_1 is the max-plus eigenvalue of $\mathbf{z}_1$, the following relation holds

$$\mathbf{z}_1(k+1) = \eta_1 \otimes \mathbf{z}_1(k). \tag{18}$$

We will show that all entries of $\boldsymbol{\eta}_i$ for $i \in \{2, \dots, p_{\max}\}$ are equal to η_1. From the property of cycle-time vector and periodic regime, we know that $\tilde{\mathbf{z}}(k+1) = \tilde{\boldsymbol{\eta}} + \tilde{\mathbf{z}}(k)$, equivalently

$$\mathbf{z}_i(k+1) = \boldsymbol{\eta}_i + \mathbf{z}_i(k), \qquad \text{for all } i \in \underline{p_{\max}}. \tag{19}$$

From (14) and (18), we obtain

$$\begin{aligned}
\mathbf{z}_i(k+1) &= \bigoplus_{j=i}^{p_{\max}} A_0^* \otimes A_j \otimes \mathbf{z}_1(k+2-j) \\
&= \bigoplus_{j=i}^{p_{\max}} A_0^* \otimes A_j \otimes \eta_1 \otimes \mathbf{z}_1(k+1-j) \\
&= \eta_1 \otimes \bigoplus_{j=i}^{p_{\max}} A_0^* \otimes A_j \otimes \mathbf{z}_1(k+1-j) \\
&= \eta_1 \otimes \mathbf{z}_i(k). \qquad (20)
\end{aligned}$$

The conclusion is derived from (19) and (20). □

The procedure to synthesize a regular schedule for public transportation networks is as follows:

1. Construct an MPL system according to Lemma 3.2;
2. Transform the MPL system to an explicit first-order model as in (13);
3. Reduce size of the model by using the reduction procedure based on Lemma 3.6;
4. Determine an entry η of the cycle-time vector and a vector $\tilde{\mathbf{d}}$ in the periodic regime by using the power algorithm (cf. Section 2.4);
5. Define $\mathbf{d}^* = [\tilde{d}_1 \quad \dots \quad \tilde{d}_n]^T$ where n represents the number of routes in the public transportation network;
6. Use the scalar η and vector $\mathbf{d}^*$ as the period and initial departure for the schedule.

Let us determine the computational complexity of the algorithm. We count the number of usual addition, usual multiplication, maximization and comparison operations in steps 1–4.

Step 1 In this step, there is only one kind of operations, namely the comparison operation. We need one comparison operation to determine an entry of matrix A_s, for a particular value of s. Thus, for each s, the number of comparison operations to construct A_s is n^2 because the size of A_s is $n \times n$. Since s ranges from 0 to $p_{\max}$, the total number of comparison operations is $(p_{\max}+1)n^2$.

Step 2 This step consists of maximization and usual addition operations. Recall that $A^* = \bigoplus_{k=0}^{n-1} A^{\otimes k}$ [15, p. 42]. Before computing A^*, we need to determine $A^{\otimes 2}$, $A^{\otimes 3}$, ..., $A^{\otimes (n-1)}$, which consist of $n-2$ max-plus matrix multiplications. For each max-plus matrix multiplication, there are $(n-1)n^2$ maximization operations and n^3 usual addition operations. Thus, the number of maximization operations is $n^2(n-1)(n-2)$ and the number of usual addition operations is $n^3(n-2)$. Then we compute the max-plus matrix addition of n matrices, where the size of each matrix is $n \times n$. In order to determine an entry, we need $n-1$ maximization operations.

Thus, the number of maximization operations is $n^2(n-1)$. The computation of $\tilde{W}$ consists of $p_{\max}$ max-plus matrix multiplications (13), i. e. $A_0^* \otimes A_1$, $A_0^* \otimes A_2$, $\ldots$, $A_0^* \otimes A_{p_{\max}}$. Notice that the size of each matrix is $n \times n$. Thus, the number of maximization operations is $p_{\max}(n-1)n^2$ and the number of usual addition operations is $p_{\max}n^3$. As a summary, the total number of maximization operations is $n^2(n-1)(n-1+p_{\max})$ and the total number of usual addition operations is $n^3(n-2+p_{\max})$.

Step 3 According to the discussion after the reduction procedure, the total number of comparison operations is $p_{\max}n(n+1)$, the total number of usual addition operations is $2p_{\max}n+1$ and the total number of usual multiplication operations is $p_{\max}$.

Step 4 In general, dimension of the model is $p_{\max}n$ (13). However, in many cases dimension of the model can be reduced (cf. Section 3.2). If $\min\{p_1, p_2, \ldots, p_n\} = 1$, we have shown that size of the reduced model is $(\sum_{u=1}^{n} p_u)$ (cf. Lemma 3.7). Since we compute the worst-case complexity, we assume dimension of the system is $p_{\max}n$. The complexity of this step is obtained by replacing the term n in the complexity of power algorithm by $p_{\max}n$. Thus, the total number of comparison operations is $(p+1)pp_{\max}n/2$, the total number of maximization operations is $(p_{\max}n-1)p_{\max}np+(p-q-1)p_{\max}n$, the total number of usual addition operations is $pp_{\max}^2n^2+1+(p_{\max}n+4)(p-q)$ and the total number of usual multiplication operations is $1+p-q$.

The worst-case complexity of the procedure for synthesizing a regular schedule is as follows. The total number of comparison operations is $n^2(p_{\max}+1)+p_{\max}n(n+1)+(p+1)pp_{\max}n/2$. The total number of maximization operations is $n^2(n-1)(n-1+p_{\max})+(p_{\max}n-1)p_{\max}np+(p-q-1)p_{\max}n$. The total number of usual addition operations is $n^3(n-2+p_{\max})+2p_{\max}n+1+pp_{\max}^2n^2+1+(p_{\max}n+4)(p-q)$. The total number of usual multiplication operations is $p_{\max}+1+p-q$.

Notice that the schedule is regular because period η is a scalar, i. e. not a vector. Furthermore the schedule for kth departure is given by $k\eta + \mathbf{d}^*$. Let us remark that η is identical to T defined in Section 2.2.

3.4. Case study

Consider the train network depicted in Figure 5. In this section, we will construct a regular schedule for the train network by using the algorithm in Section 3.3. The max-plus matrix representing the transportation network is given by

$$N = \begin{bmatrix} 3 & 6 \\ 4 & 5 \end{bmatrix}$$

Notice that there are two stations {station 1, station 2} and four routes $\{e_1 = (1,1)$, $e_2 = (1,2)$, $e_3 = (2,2)$, $e_4 = (2,1)\}$. Thus, $m = 2$ and $n = 4$. The initial number of public vehicles in all routes is $p_1 = 1$, $p_2 = 2$, $p_3 = 1$ and $p_4 = 0$.

Let us construct an implicit higher-order MPL system by using Lemma 3.2. First, we focus on the dynamics of state x_1 that corresponds to route e_1. Thus we restrict ourself

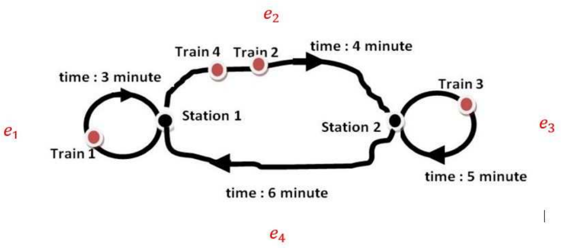

Fig. 5. A simple train network.

on the first row. Since $p_1 = 1$, we can restrict our attention to matrix A_1. The origin of route e_1 is station 1. Thus we have to find all routes such that the destination is station 1. Those routes are e_1 and e_4. Finally, we define $A_1(1,1)$ as the traveling time of route e_1 and $A_1(1,4)$ as the traveling time of route e_4. The dynamics of other states can be determined similarly. The resulting implicit higher-order MPL system is

$$\mathbf{x}(k+1) = A_0 \otimes \mathbf{x}(k+1) \oplus A_1 \otimes \mathbf{x}(k) \oplus A_2 \otimes \mathbf{x}(k-1),$$

where

$$A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & 5 & \varepsilon \end{bmatrix}, \quad A_1 = \begin{bmatrix} 3 & \varepsilon & \varepsilon & 6 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & 5 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \quad A_2 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon & 6 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}.$$

Next we transform the above system to an explicit higher-order MPL system. In order to do that, we need to compute A_0^*. Since $A_0^{\otimes 2} = \mathcal{E}_{4,4}$, it follows that

$$A_0^* = E_4 \oplus A_0 = \begin{bmatrix} 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 \end{bmatrix} \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & 5 & \varepsilon \end{bmatrix} = \begin{bmatrix} 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & 4 & 5 & 0 \end{bmatrix}.$$

The explicit higher-order MPL system is given by

$$\mathbf{x}(k+1) = A_0^* \otimes A_1 \otimes \mathbf{x}(k) \oplus A_0^* \otimes A_2 \otimes \mathbf{x}(k-1),$$

where

$$A_0^* \otimes A_1 = \begin{bmatrix} 3 & \varepsilon & \varepsilon & 6 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & 5 & \varepsilon \\ \varepsilon & 9 & 10 & \varepsilon \end{bmatrix}, \quad A_0^* \otimes A_2 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon & 6 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 7 & \varepsilon & \varepsilon & 10 \end{bmatrix}.$$

Then we construct an explicit first-order MPL system as discussed in the end of

Section 3.1, as follows

$$\tilde{\mathbf{z}}(k+1) = \begin{bmatrix} A_0^* \otimes A_1 & E_4 \\ A_0^* \otimes A_2 & \mathcal{E}_{4,4} \end{bmatrix} \otimes \tilde{\mathbf{z}}(k)$$
$$= \begin{bmatrix} 3 & \varepsilon & \varepsilon & 6 & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & 4 & 5 & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & 9 & 10 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 7 & \varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \otimes \tilde{\mathbf{z}}(k).$$

According to Lemma 3.6, we can remove the fifth and seventh rows because all entries in those rows are infinite. The reduced MPL system is

$$\begin{bmatrix} \tilde{z}_1(k+1) \\ \tilde{z}_2(k+1) \\ \tilde{z}_3(k+1) \\ \tilde{z}_4(k+1) \\ \tilde{z}_6(k+1) \\ \tilde{z}_8(k+1) \end{bmatrix} = \begin{bmatrix} 3 & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & 4 & 5 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 9 & 10 & \varepsilon & \varepsilon & 0 \\ 3 & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon \\ 7 & \varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} \tilde{z}_1(k) \\ \tilde{z}_2(k) \\ \tilde{z}_3(k) \\ \tilde{z}_4(k) \\ \tilde{z}_6(k) \\ \tilde{z}_8(k) \end{bmatrix}.$$

Next we determine an entry of the cycle-time vector η and a state $\tilde{\mathbf{d}}$ of the preceding state matrix by using the power algorithm (cf. Section 2.4). We obtain $\eta = 5$ and $\tilde{\mathbf{d}} = [6 \quad 1 \quad 0 \quad 5 \quad 6 \quad 10]^T$. As mentioned in the algorithm in Section 3.3, initial schedule $\mathbf{d}^*$ is defined as $[6 \quad 1 \quad 0 \quad 5]^T$. The regular schedule for the first few train departures can be seen in Table 1.

kth departure	S_1 to S_1	S_1 to S_2	S_2 to S_2	S_2 to S_1
1	06.06	06.01	06.00	06.05
2	06.11	06.06	06.05	06.10
3	06.16	06.11	06.10	06.15
4	06.21	06.16	06.15	06.20
5	06.26	06.21	06.20	06.25

Tab. 1. Schedule of the first 5 departures, where the earliest departure is 06.00 and the unit for traveling time is minute.

4. CONCLUSIONS

We have derived an algorithm to construct a regular schedule for public transportation network by using max-plus algebra. The network is assumed to be strongly connected. The input of this algorithm is the graph of the public transportation network, the traveling time of each route and the number of public vehicles serving each route. We have proposed a novel technique to transform an explicit higher-order MPL system into an explicit first-order MPL system. This technique allows us to construct a systematic procedure to reduce the MPL system. Furthermore, we have used the so-called power algorithm to determine a period and initial departure of the regular schedule of transportation systems.

(Received October 16, 2016)

REFERENCES

[1] P. Arnold, D. Peeters, and I. Thomas: Modelling a rail/road intermodal transportation system. Transport. Res. Part E: Logist. Transport. Rev. *40* (2004), 255–270. DOI:10.1016/j.tre.2003.08.005

[2] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat: Synchronization and Linearity, an Algebra for Discrete Event Systems. John Wiley and Sons, 1992.

[3] J. G. Braker: Algorithms and Applications in Timed Discrete Event Systems. PhD Thesis, Delft University of Technology, 1993.

[4] L. Castelli, R. Pesenti, and W. Ukovich: Scheduling multimodal transportation systems. Europ. J. Oper. Res. *155* (2004), 603–615. DOI:10.1016/j.ejor.2003.02.002

[5] B. Chen and H. H. Cheng: A review of the applications of agent technology in traffic and transportation systems. IEEE Trans. Intell. Transport. Systems *11* (2010), 485–497. DOI:10.1109/tits.2010.2048313

[6] J. Cochet-Terrasson, G. Cohen, S. Gaubert, M. McGettrick, and J.-P. Quadrat: Numerical computation of spectral elements in max-plus algebra. In: Proc. IFAC Conference on System Structure and Control, 1998, pp. 699–706. DOI:10.1016/s1474-6670(17)42067-2

[7] T. G. Crainic and J.-M. Rousseau: Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. Transport. Res. Part B: Methodological *20* (1986), 225–242. DOI:10.1016/0191-2615(86)90019-6

[8] A. Di Febbraro and S. Sacone: Hybrid modelling of transportation systems by means of Petri nets. In: Proc. IEEE International Conference on Systems, Man, and Cybernetics, 1998, pp. 131–135. DOI:10.1109/icsmc.1998.725397

[9] A. Di Febbraro and N. Sacco: On modelling urban transportation networks via hybrid Petri nets. Control Engrg. Practice *12* (2004), 1225–1239. DOI:10.1016/j.conengprac.2004.04.008

[10] M. M. Etschmaier: Fuzzy controls for maintenance scheduling in transportation systems. Automatica *16* (1980), 255–264. DOI:10.1016/0005-1098(80)90035-7

[11] K. Fahim, L. Hanafi, and F. Ayu: Monorail and tram scheduling which integrated Surabaya using max-plus algebra. In: International Seminar on Innovation in Mathematics and Mathematics Education, 2014.

[12] K. Fahim, Subiono, and J. W. van der Woude: On a generalization of power algorithms over max-plus algebra. Discrete Event Dynamic Systems *27* (2017), 181–203. DOI:10.1007/s10626-016-0235-4

[13] R. M. P. Goverde: Punctuality of Railway Operations and Timetable Stability Analysis. PhD Thesis, Delft University of Technology, 2005.

[14] B. B. Gursoy and O. Mason: Spectral properties of matrix polynomials in the max algebra. Linear Algebra Appl. *435* (2011), 1626–1636. DOI:10.1016/j.laa.2010.01.014

[15] B. Heidergott, G. J. Olsder, and J.W. van der Woude: Max Plus at Work–Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications. Princeton University Press, 2006. DOI:10.1515/9781400865239

[16] D. Herrero-Perez and H. Martinez-Barbera: Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems. IEEE Trans. Industrial Informatics *6* (2010), 166–180. DOI:10.1109/tii.2009.2038691

[17] S. J. James, C. James, and J. A. Evans: Modelling of food transportation systems – a review. Int. J. Refrigeration *29* (2006), 947–957. DOI:10.1016/j.ijrefrig.2006.03.017

[18] A. Levin: Scheduling and fleet routing models for transportation systems. Transport. Sci. *5* (1971), 232–255. DOI:10.1287/trsc.5.3.232

[19] G. Soto y Koelemeijer: On the Behaviour of Classes of Min-max-plus Systems. PhD Thesis, Delft University of Technology, 2003.

[20] D. M. Stein: Scheduling dial-a-ride transportation systems. Transport. Sci. *12* (1978), 232–249. DOI:10.1287/trsc.12.3.232

[21] Subiono and J.W. van der Woude: Power algorithms for (max,+)- and bipartite (min,max,+)-systems. Discrete Event Dynamic Systems *10* (2000), 369–389. DOI:10.1023/a:1008315821604

[22] Subiono: On Classes of Min-max-plus Systems and their Applications. PhD Thesis, Delft University of Technology, 2000.

[23] Subiono and K. Fahim: On computing supply chain scheduling using max-plus algebra. Appl. Math. Sci. *10* (2016), 477–486. DOI:10.12988/ams.2016.618

Subiono, Department of Mathematics, Faculty of Mathematics, Computing, and Data Science, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo-Surabaya 60111, Indonesia
e-mail: subiono2008@matematika.its.ac.id

Kistosil Fahim, Department of Mathematics, Faculty of Mathematics, Computing, and Data Science, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo-Surabaya 60111, Indonesia
e-mail: kfahim@matematika.its.ac.id

Dieky Adzkiya, Department of Mathematics, Faculty of Mathematics, Computing, and Data Science, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo-Surabaya 60111, Indonesia
e-mail: dieky@matematika.its.ac.id