symmetry

MDPI

Article

An Efficient Algorithm for Nontrivial Eigenvectors in Max-Plus Algebra

Mubasher Umer [1], Umar Hayat [1,*] and Fazal Abbas [2,*]

[1] Department of Mathematics, Quaid-i-Azam University, Islamabad 45320, Pakistan; mumer@math.qau.edu.pk

[2] Department of Mathematics and Computer Sciences, Stetson University, DeLand, FL 32723, USA

[*] Correspondence: umar.hayat@qau.edu.pk (U.H.); fabbas1@stetson.edu (F.A.); Tel.: +92-51-90642192 (U.H.); +1-585-993-4291 (F.A.)

Received: 30 April 2019; Accepted: 27 May 2019; Published: 30 May 2019

check for updates

Abstract: The eigenproblem for matrices in max-plus algebra describes the steady state of the system, and therefore it has been intensively studied by many authors. In this paper, we propose an algorithm to compute the eigenvalue and the corresponding eigenvectors of a square matrix in an iterative way. The algorithm is extended to compute the nontrivial eigenvectors for Latin squares in max-plus algebra.

Keywords: max-plus algebra; Latin squares; eigenvalue and eigenvectors

1. Introduction

Discrete event systems can be used to study processes that are driven by the occurrence of events. The relevant variables represent times at which events are taking place. An important class of systems consists of min-max-plus systems in which the operations—the maximization, the minimization and the addition, can describe discrete event dynamic systems. The union of two sets of equations describes the min-max-plus systems, where the maximization and addition are contained in one set of equations and the minimization and addition are contained in another set of equations.

Max-plus algebra $\mathbb{R}_\epsilon = \mathbb{R} \cup \{\epsilon\}$ where $\epsilon = -\infty$, is a set which deals with two main operations—$\oplus$ as maximization and $\otimes$ as addition. These operations endow $\mathbb{R}_\epsilon$ with the structure of a commutative semiring.

There are various applications of max-plus algebra in different areas of mathematics [1]. Max-plus algebra is also considered in traffic control, parallel processing systems, telecommunication networks, manufacturing systems, machine scheduling, and control theory [2–4]. In [5], the author has described the model of all train connections of the whole Dutch railway system with all train types by a max-plus system. Max-plus algebra can be used to model disk events related to synchronization and time delays [6]. A new steganography method has been proposed to hide an image into another image using matrix multiplication operations on max-plus algebra [7]. In [8] the author has shown how max-plus algebra can be helpful for the dynamic programming of algorithms. The primary reason for the utilization of max-plus algebra in different fields is that many equations which are non-linear in conventional algebra become linear in max-plus algebra [3].

In this paper, we consider the eigenproblems in max-plus algebra. Eigenproblems are the problems of finding an eigenvalue λ and an eigenvector v from an $n \times n$ matrix M such that $Mv = \lambda v$. The algorithm in [9] can solve these problems.

A novel iterative method is proposed for the computation of eigenvalue and the corresponding eigenvectors in max-plus algebra. To evaluate the eigenvalue and the corresponding eigenvectors of discrete event systems, some algorithms have been established in [8–11]. The first algorithm to determine the eigenvalue was given by Karp [12]. A new algorithm was proposed to solve systems of max-plus linear inequalities [13]. This algorithm uses the technique of cyclic projectors. In [14], there is a method for determining the eigenvectors of monotone and homogenous functions.

The paper is structured as follows. In Section 2, basic results and definitions related to max-plus algebra are described. A new algorithm is proposed to determine the eigenvalue and eigenvectors in an iterative way. In Section 3, we discuss the Latin squares in max-plus algebra and extend the algorithm for Latin squares. Conclusions are made in section 4.

2. Max-Plus Algebra

In this section, we recall some basic results and definitions related to the max-plus algebra. The symbols $(\mathbb{R}_\epsilon, \oplus, \otimes)$ denote the algebraic structure of max-plus algebra where $\mathbb{R}_\epsilon$ is the set $\mathbb{R} \cup \{-\infty\}$ and $\otimes, \oplus$ are the binary operations defined as

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

and

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

These operations endow $\mathbb{R}_\epsilon$ with a structure of commutative semiring. Both the operations $\otimes$ and $\oplus$ satisfy associativity and commutativity.

Also for all $m \in \mathbb{R}_\epsilon$

$$\begin{aligned} m \oplus \epsilon = \epsilon \oplus m = m, \\ m \otimes 0 = 0 \otimes m = m. \end{aligned}$$

So in max-plus algebra, ϵ is a null element and 0 is a unit element.

For $m \in \mathbb{R}_\epsilon$ and a non-negative integer n, define

$$m^{\otimes n} = \begin{cases} \underbrace{m \otimes m \otimes m \otimes ... \otimes m}_{n\text{-}times} & \text{for } n > 0, \\ 0 & \text{for } n = 0. \end{cases}$$

Using the standard multiplication in $\mathbb{R}$, we have

$$m^{\otimes n} = nm.$$

In max-plus algebra, $\mathbb{R}_\epsilon^{m \times n}$ and $\mathbb{R}_\epsilon^{m}$ denote the set of all matrices of order $m \times n$ with entries in $\mathbb{R}_\epsilon$, and the set of all $m \times 1$ vectors respectively. The set of first n natural numbers is denoted by $\underline{n}$, *i.e.*, $\underline{n} = \{1, ..., n\}$. Now let $M = (m_{ij})$ and $N = (n_{ij})$ be elements of $\mathbb{R}_\epsilon^{m \times n}$. The sum of these two matrices $M \oplus N = T = (t_{ij})$ is defined as

$$t_{ij} = m_{ij} \oplus n_{ij} = max(m_{ij}, n_{ij}) \quad \text{for } i, j \in \underline{n}.$$

Let $M \in \mathbb{R}_\epsilon^{m \times k}$ and $N \in \mathbb{R}_\epsilon^{k \times n}$ be two matrices. The product $M \otimes N = T \in \mathbb{R}_\epsilon^{m \times n}$ is defined by the formula

$$t_{ij} = \bigoplus_{l=1}^{k} (m_{il} \otimes n_{lj}) = \max_{l \in \underline{k}} (m_{il} + n_{lj}) \quad \text{for } \ i \in \underline{m}; \ \ j \in \underline{n}.$$

For $M \in \mathbb{R}_\epsilon^{n \times n}$, the *kth* power of M can be defined as

$$M^{\otimes k} = M \otimes M^{\otimes k-1}.$$

The starting point is the following system of equations describing the time evolution of a discrete event system:

$$u_i(k) = \underset{1\le j\le n}{max}\{m_{ij} + u_j(k-1)\} \quad \text{for all} \;\; i \in \{1, ..., n\} \;\; \text{and} \;\; k \in \mathbb{N} \setminus \{0\}.$$

This can be written as

$$u_i(k) = \bigoplus_{j=1}^{n}(m_{ij} \otimes u_j(k-1)).$$

Finally, the previous equation can compactly be written as

$$u(k) = M \otimes u(k-1), \tag{1}$$

where $u(k-1) = [u_1(k-1), ..., u_n(k-1)]^T \in \mathbb{R}^n_\epsilon$, $M \in \mathbb{R}^{n\times n}_\epsilon$.

System (1) has an eigenvalue $\lambda \in \mathbb{R}$ and the corresponding eigenvector $v \in \mathbb{R}^n$ if

$$M \otimes v = \lambda \otimes v. \tag{2}$$

For each matrix $M \in \mathbb{R}^{n\times n}_\epsilon$, we can construct a directed graph (digraph) $\mathcal{G}(M) = \mathcal{G}(N, E)$, where N is the set of vertices and E represents the set of edges. There exists an arc $i \to j$ if $m_{ji} \neq -\infty$; such an arc will be denoted by (i, j). The weight $\omega(j, i)$ for each (i, j)-arc is defined as $\omega(j, i) = m_{ji}$. If $m_{ji} = -\infty$, then there is no (i, j)-arc. A path is a sequence of arcs $(a_1, a_2), ..., (a_{k-1}, a_k)$. If all vertices $a_1, ..., a_k$ are different then the path is called an elementary path. An elementary path of the form $(a_1, a_2), ..., (a_{k-1}, a_1)$ is known as a circuit. A loop is a circuit consisting of an edge that connects a vertex to itself. The weight of a path $p = (a_1, a_2), ..., (a_{k-1}, a_k)$, denoted by $\omega(p)$, is defined as the sum of weights of each arc. The length of p, denoted by $l(p)$, is equal to the number of arcs in the path p. The average weight of a path p is defined by $\dfrac{\omega(p)}{l(p)}$. A critical circuit is a circuit with maximum average weight. If there exists a path from each vertex i to each vertex j then the graph is said to be strongly connected. A matrix M is irreducible if and only if the digraph of M is strongly connected [15].

The eigenvalue of a matrix M is equal to the maximal average weight in $\mathcal{G}(M)$ [16]. The eigenvalue for an irreducible matrix M, if it exists, is unique [16]. So, problem of getting the eigenvalue of a matrix M related to the problem of getting the critical circuits in $\mathcal{G}(M)$.

In this paper, we find the eigenvalue $\lambda \in \mathbb{R}$ of a matrix $M \in \mathbb{R}^{n\times n}_\epsilon$ with the help of the maximal average weight in $\mathcal{G}(M)$ as done in [16]. Then we define the matrix M_λ as $M_\lambda = -\lambda \otimes M$, in other words $[M_\lambda]_{i,j} = [M]_{i,j} - \lambda$. At the end, we define

$$u^*(k) = M_\lambda \otimes u^*(k-1), \tag{3}$$

for $k \in \mathbb{N} \setminus \{0\}$, where $u^*(k-1) = [u^*_1(k-1), ..., u^*_n(k-1)]^T \in \mathbb{R}^n_\epsilon$. Using this relation, an algorithm is derived to calculate the associated eigenvectors of a matrix M.

Theorem 1. *Let $M \in \mathbb{R}^{n\times n}_\epsilon$ be a matrix and let*

$$u^*(k) = M_\lambda \otimes u^*(k-1),$$

for $k \in \mathbb{N} \setminus \{0\}$, where $u^*(0) \neq \epsilon$. If there exist some positive integers $r > s \geq 0$, such that $u^*(r) = u^*(s)$, then

$$M \otimes v = \lambda \otimes v,$$

where

$$v = u^*(s) \oplus ... \oplus u^*(r-1).$$

Proof of Theorem 1. The main task is to prove that $M \otimes v = \lambda \otimes v$. Indeed, we have

$$\begin{aligned} M \otimes v &= \lambda \otimes (-\lambda) \otimes M \otimes v \\ &= \lambda \otimes M_\lambda \otimes v \\ &= \lambda \otimes M_\lambda \otimes \{u^*(s) \oplus ... \oplus u^*(r-1)\} \\ &= \lambda \otimes \{M_\lambda \otimes u^*(s) \oplus ... \oplus M_\lambda \otimes u^*(r-1)\} \\ &= \lambda \otimes \{u^*(s+1) \oplus ... \oplus u^*(r)\} \\ &= \lambda \otimes v. \end{aligned}$$

□

The main algorithm of this article stated below (Algorithm 1) gives the eigenvalue and the corresponding eigenvectors. This algorithm consists of the following steps:

Algorithm 1 Eigenproblems in Max-Plus Algebra

```
1. Compute the maximal circuit mean $\lambda$, in $\mathcal{G}(M)$.
2. Define $M_\lambda = -\lambda \otimes M$.
3. Define an arbitrary initial vector $u^*(0) \neq \epsilon$.
4. Iterate $u^*(k) = M_\lambda \otimes u^*(k-1)$, for $k \in \mathbb{N} \setminus \{0\}$, until there are positive integers $r > s \geq 0$, such
   that $u^*(r) = u^*(s)$.
5. Compute the eigenvector
       $v = u^*(s) \oplus ... \oplus u^*(r-1).$
```

Each step of the above algorithm is explained with the help of the following example.

Example 1. *Consider the following matrix* $M \in \mathbb{R}_\epsilon^{4\times 4}$

$$M = \begin{bmatrix} \epsilon & 5 & 2 & 2 \\ 3 & \epsilon & 1 & \epsilon \\ 2 & 1 & 3 & \epsilon \\ \epsilon & 2 & 2 & \epsilon \end{bmatrix}.$$

The corresponding graph $\mathcal{G}(M)$ *is depicted in Figure 1. Since the average weight of the circuit* $1 \to 2 \to 1$ *is the maximal circuit mean, the eigenvalue* $\lambda = 4$. *Now we get*

$$M_\lambda = \begin{bmatrix} \epsilon & 1 & -2 & -2 \\ -1 & \epsilon & -3 & \epsilon \\ -2 & -3 & -1 & \epsilon \\ \epsilon & -2 & -2 & \epsilon \end{bmatrix}.$$

Now take the initial vector

$$u^*(0) = \begin{pmatrix} 0 \\ \epsilon \\ \epsilon \\ \epsilon \end{pmatrix}.$$

The following sequence is obtained by iterating (3),

$$\begin{array}{ccccccccc} u^*(1) & \to & u^*(2) & \to & u^*(3) & \to & u^*(4) & = u^*(2) \\ \begin{pmatrix} \epsilon \\ -1 \\ -2 \\ \epsilon \end{pmatrix} & \to & \begin{pmatrix} 0 \\ -5 \\ -3 \\ -3 \end{pmatrix} & \to & \begin{pmatrix} -4 \\ -1 \\ -2 \\ -5 \end{pmatrix} & \to & \begin{pmatrix} 0 \\ -5 \\ -3 \\ -3 \end{pmatrix} & = u^*(2). \end{array}$$

It follows that $s = 2, r = 4$*. We proceed with the computation of the vector* v*.*

$$\begin{aligned} v &= u^*(2) \oplus u^*(3) \\ &= \begin{pmatrix} 0 \\ -5 \\ -3 \\ -3 \end{pmatrix} \oplus \begin{pmatrix} -4 \\ -1 \\ -2 \\ -5 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ -2 \\ -3 \end{pmatrix}. \end{aligned}$$

It is easy to see that indeed

$$M \otimes v = \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \end{pmatrix} = \lambda \otimes v.$$

Hence for the eigenvalue $\lambda = 4$*, the corresponding eigenvector* v *of the matrix* M *resulting from the above algorithm is a correct eigenvector.*

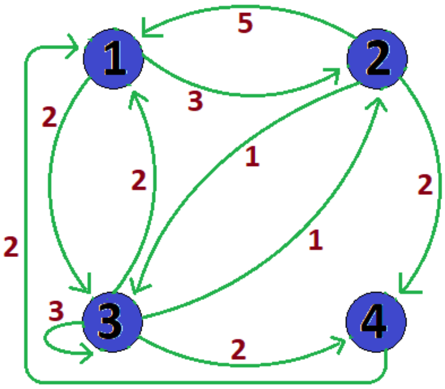

Figure 1. Graph representation of matrix M.

3. Latin Square

In this section, the proposed algorithm is used to compute the eigenvalue and the corresponding eigenvectors of Latin squares. A Latin square is a square matrix of size $n \times n$ with n different numbers such that each row and each column contains distinct numbers [17]. An example of a Latin square is given in the following

$$L = \begin{pmatrix} 3 & 2 & 4 & 1 \\ 4 & 1 & 3 & 2 \\ 2 & 4 & 1 & 3 \\ 1 & 3 & 2 & 4 \end{pmatrix}.$$

The purpose behind considering eigenproblems for Latin squares in max-plus algebra is that similar issues have been examined for different matrices like circulant matrices [18,19], Monge matrices [20] and inverse Monge matrices [21]. Eigenproblems for these special matrices are more easy to answer. For example, the maximal entry in circulant matrices is its eigenvalue.

Now we will discuss two different types of Latin square.

• The entries of L are $\underline{n} = \{1, ..., n\}$.

• The entries of L are $\underline{n}_\epsilon = \{1, ..., n-1, \epsilon\}$.

Let L be a Latin square with entries in $\underline{n}$. Since each entry in L is finite, the corresponding graph $\mathcal{G}(L)$ is strongly connected. Similarly, for a Latin square L with entries in $\underline{n}_\epsilon$, exactly one infinite entry appears in each row and column. So there exists a path of length 2 from each vertex r to each vertex s

in $\mathcal{G}(L)$. Consequently, the graph $\mathcal{G}(L)$ is strongly connected. Hence in both cases, a Latin square is an irreducible matrix. In [11] authors show that for a Latin square L, the maximal number in L is equal to the maximal circuit mean in $\mathcal{G}(L)$. Therefore the eigenvalue of a Latin square L is equal to the maximal number in L [11].

Now using this concept, the proposed algorithm (Algorithm 2) computes the eigenvalue and eigenvectors of Latin square as follows:

Algorithm 2 Eigenproblems for Latin Square

```
1. Compute the eigenvalue $\lambda$ as the maximal number in $L$.
2. Define $M_\lambda = -\lambda \otimes M$.
3. Define an arbitrary initial vector $u^*(0) \neq \epsilon$.
4. Iterate $u^*(k) = M_\lambda \otimes u^*(k-1)$, for $k \in \mathbb{N} \setminus \{0\}$, until there are positive integers $r > s \geq 0$, such that $u^*(r) = u^*(s)$.
5. Compute the eigenvector
       $v = u^*(s) \oplus ... \oplus u^*(r-1).$
```

Also, recall the power algorithm presented by Subiono [9]. This algorithm computes the eigenvalue and corresponding eigenvector for max-plus systems of type (1).

Now we illustrate the Algorithms 2 and 3 by means of an example. In the following example, based on Algorithms 2 and 3, M is a Latin square with entries in $\underline{n}_\epsilon$.

Algorithm 3 Eigenproblems for Max-Plus Algebra

```
1. Take an arbitrary initial state vector $u(0) \neq \epsilon$.
2. Iterate (1) until there are integers $p; q$ with $p > q \geq 0$ and a real number c, such that $u(p) = c \otimes u(q)$.
3. Define as eigenvalue $\lambda = c/p - q$ (in the conventional sense).
4. Define as eigenvector $v = \oplus_{j=1}^{p-q}(\lambda^{\otimes(p-q-j)} \otimes u(q+j-1))$.
```

Example 2. *Consider a Latin square*

$$M = \begin{bmatrix} 2 & 1 & 3 & \epsilon \\ 1 & \epsilon & 2 & 3 \\ 3 & 2 & \epsilon & 1 \\ \epsilon & 3 & 1 & 2 \end{bmatrix}.$$

1. *By Algorithm 2, the maximal number in M is the eigenvalue, i.e.,*

$$\lambda = max(M) = 3.$$

Consequently,

$$M_\lambda = \begin{bmatrix} -1 & -2 & 0 & \epsilon \\ -2 & \epsilon & -1 & 0 \\ 0 & -1 & \epsilon & -2 \\ \epsilon & 0 & -2 & -1 \end{bmatrix}.$$

Next, consider the initial vector

$$u^*(0) = \begin{pmatrix} 0 \\ \epsilon \\ \epsilon \\ \epsilon \end{pmatrix}.$$

After iterating (3), the following sequence is obtained:

$$\begin{array}{ccccccccc} u^*(1) & \rightarrow & u^*(2) & \rightarrow & u^*(3) & \rightarrow & u^*(4) & = u^*(2) \\ \begin{pmatrix} -1 \\ -2 \\ 0 \\ \epsilon \end{pmatrix} & \rightarrow & \begin{pmatrix} 0 \\ -1 \\ -1 \\ -2 \end{pmatrix} & \rightarrow & \begin{pmatrix} -1 \\ -2 \\ 0 \\ -1 \end{pmatrix} & \rightarrow & \begin{pmatrix} 0 \\ -1 \\ -1 \\ -2 \end{pmatrix} & = u^*(2). \end{array}$$

It follows that $s = 2, r = 4$. To compute a corresponding eigenvector we proceed with the computation of the vector v.

$$\begin{aligned} v &= u^*(s) \oplus ... \oplus u^*(r-1) \\ &= u^*(2) \quad \oplus \quad u^*(3) \\ &= \begin{pmatrix} 0 \\ -1 \\ -1 \\ -2 \end{pmatrix} \oplus \begin{pmatrix} -1 \\ -2 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 0 \\ -1 \end{pmatrix}. \end{aligned}$$

It is easy to check whether or not $M \otimes v = \lambda \otimes v$. It follows that

$$M \otimes v = \begin{pmatrix} 3 \\ 2 \\ 3 \\ 2 \end{pmatrix} = \lambda \otimes v.$$

Hence for the eigenvalue $\lambda = 3$, the corresponding eigenvector v of the matrix M resulting from the above algorithm is a correct eigenvector.

2. *For Algorithm 3, consider the initial vector*

$$u(0) = \begin{pmatrix} 0 \\ \epsilon \\ \epsilon \\ \epsilon \end{pmatrix}.$$

Iterating (1), we obtain

$$\begin{array}{ccccccccc} u(1) & \rightarrow & u(2) & \rightarrow & u(3) & \rightarrow & u(4) & = u(2) \\ \begin{pmatrix} 2 \\ 1 \\ 3 \\ \epsilon \end{pmatrix} & \rightarrow & \begin{pmatrix} 6 \\ 5 \\ 5 \\ 4 \end{pmatrix} & \rightarrow & \begin{pmatrix} 8 \\ 7 \\ 9 \\ 8 \end{pmatrix} & \rightarrow & \begin{pmatrix} 12 \\ 11 \\ 11 \\ 12 \end{pmatrix} & = 6 \otimes u(2). \end{array}$$

It follows that $q = 2, p = 4$. The vector v resulting from Algorithm 3, is given as

$$\begin{aligned} v &= \oplus_{j=1}^{p-q}(\lambda^{\otimes(p-q-j)} \otimes u(q+j-1)) \\ &= \oplus_{j=1}^{2}(\lambda^{\otimes(2-j)} \otimes u(2+j-1)) \\ &= \lambda \otimes u(2) \;\; \oplus \;\; u(3) \\ &= 3 \otimes \begin{pmatrix} 6 \\ 5 \\ 5 \\ 4 \end{pmatrix} \oplus \begin{pmatrix} 8 \\ 7 \\ 9 \\ 8 \end{pmatrix} \\ &= \begin{pmatrix} 9 \\ 8 \\ 8 \\ 7 \end{pmatrix} \oplus \begin{pmatrix} 8 \\ 7 \\ 9 \\ 8 \end{pmatrix} = \begin{pmatrix} 9 \\ 8 \\ 9 \\ 8 \end{pmatrix}. \end{aligned}$$

Which is the correct eigenvector.

Remark 1. *The main purpose of this article is to present an alternative algorithm for the computation of an eigenvalue and the associated eigenvector in max-plus algebra. This algorithm works in an iterative way. Here we give a computational comparison of the Algorithm 2 with the Algorithm 3 given by Subiono and van dar Woude [9]. In the case of a circulant matrix or a Latin square, the maximum value is the eigenvalue. Therefore the Algorithm 2 works quite easily compared to the Algorithm 3 This is because Algorithm 3 ends up in a periodic behavior if there exists a real number c and integers $r > s \geq 0$, such that $u(r) = c \otimes u(s)$, while in the case of Algorithm 2, we do not need any real number c. Also the Algorithm 3 gives an eigenvector v as*

$$v = \oplus_{j=1}^{r-s}(\lambda^{\otimes(r-s-j)} \otimes u(s+j-1)),$$

however, when using the Algorithm 2 one gets an eigenvector v as $v = u^*(s) \oplus ... \oplus u^*(r-1)$. Hence the computation of an eigenvector using the Algorithm 2 is quite easy compared to the Algorithm 3.

4. Conclusions

In this article, the eigenproblem for max-plus algebra is discussed. A detailed proof has been presented for the proposed algorithm that can calculate the eigenvalue and the corresponding eigenvectors of strongly connected max-plus algebra systems. The algorithm is implemented by means of a simple example. In addition, this algorithm has been extended for Latin squares in max-plus algebra and compared with other existing algorithms. In the future, using the idea of this article, an algorithm can be derived to compute the eigenvalue and an associated eigenvector for bipartite (min, max, +)-systems.

Author Contributions: All authors contributed equally to this work.

Funding: Fazal Abbas acknowledges the financial support provided by the College of Arts and Sciences, Stetson University, USA.

Acknowledgments: We thank the referees for their valuable suggestions and helpful remarks.

Conflicts of Interest: The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Halburd, R.G.; Southall, N.J. Tropical nevanlinna theory and ultra-discrete equations. *Int. Math. Res. Not.* **2009**, *5*, 887–911.
2. Cuninghame-Green, R.A. Lecture notes in economics and mathematical systems. In *Minimax Algebra*; Springer-Verlag: New York, NY, USA, 1979.
3. De Shutter, B. On the ultimate behavior of the sequence of consecutive powers of a matrix in the max-plus algebra. *Linear Algebra Its Appl.* **2000**, *307*, 103–117. [CrossRef]
4. Gaubert, S. Methods and applications of (max,+) linear algebra. In *Annual Symposium on Theoretical Aspects of Computer Science*; Springer-Verlag: Berlin/Heidelberg, Germany, 1997; pp. 261–282.
5. Subiono on Classes of Min-Max-Plus Systems and Their Application. Ph.D. Thesis, Delft University of Technology, Delft , The Netherlands, 2000.
6. Kubo, S.; Nishinari, K. Applications of max-plus algebra to flow shop scheduling problems. *Discret. Appl. Math.* **2018**, *247*, 278–293. [CrossRef]
7. Santoso, K.A.; Suprajitno, H. On Max-Plus Algebra and Its Application on Image Steganography. *Sci. World J.* **2018**, *2018*, 6718653. [CrossRef] [PubMed]
8. Braker, J.G.; Olsder, G.J. The power algorithm in max algebra. *Linear Algebra Its Appl.* **1993**, *182*, 67–89. [CrossRef]
9. Subiono; van der Woude, J. Power algorithms for (max,+)- and bipartite (min,max,+)-systems. *Discret. Event Dyn. Syst.* **2000**, *10*, 369–389. [CrossRef]
10. Garca-Planas, M.I.; Magret, M.D. Eigenvectors of permutation matrices. *Adv. Pure Math.* **2015**, *5*, 390–394. [CrossRef]
11. Mufid, M.S.U. Subiono Eigenvalues and eigenvectors of latin squares in max-plus algebra. *J. Indones. Math. Soc.* **2014**, *20*, 37–45. [CrossRef]
12. Karp, R.M. A characterization of the minimum cycle mean in a digraph. *Discret. Math.* **1978**, *23*, 309–311. [CrossRef]
13. Akian, M.; Gaubert, S.; Nitica, V.; Singer, I. Best approximation in maxplus semimodules. *Linear Algebra Its Appl.* **2011**, *435*, 3261–3296. [CrossRef]
14. Gunawardena, J. Min-max functions. *Discret. Event Dyn. Syst.* **1994**, *4*, 377–407. [CrossRef]
15. Brualdi, R.A.; Ryser, H.J. *Combinatorial Matrix Theory*; Cambridge University Press: Cambridge, UK, 1991.
16. Subiono, A.M. Aljabar max-plus dan terapannya. Buku Ajar Mata Kuliah Pilihan Pasca Sarjana Matematika. In Proceedings of the National Graduate Seminar XII—ITS, Surabaya, Indonesia, 11–12 October 2012.
17. McKay, B.D.; Wanless, I.M. On the number of Latin squares. *Ann. Comb.* **2005**, *9*, 334–344. [CrossRef]
18. Tomaskova, H. Eigenproblem for circulant matrices in max-plus algebra. In Proceedings of the 12th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems, Kantaoui, Sousse, Tunisia, 3–6 May 2010; World Scientific and Engineering Academy and Society (WSEAS): Cambridge, UK, 2010; pp. 158–161.
19. Ulfa, M. Analysis on Eigenvector of Circulant Matrix in Max-Plus Algebra. Master's Thesis, Mathematics and Natural Science Faculty, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, 2009. (In Indonesian)
20. Gavalec, M.; Plavka, J. Structure of the eigenspace of a Monge matrix in max-plus algebra. *Discret. Appl. Math.* **2008**, *10*, 596–606. [CrossRef]
21. Imaev, A.A.; Judd, R.P. Computing an eigenvector of an inverse Monge matrix in maxplus algebra. *Discret. Appl. Math.* **2010**, *158*, 1701–1707. [CrossRef]