Table of Contents

MAX-PLUS ALGEBRA:
PROPERTIES AND APPLICATIONS

by Maria H. Andersen

A thesis submitted to the Department of Mathematics in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE in MATHEMATICS.

Laramie, WY
May 2002

Introduction

The goal of this paper is to describe a mathematical theory, called the max-plus algebra, which affords a uniform treatment of many problems that arise in the field of Operations Research. We illustrate several applications of this theory with detailed examples from transportation networking, project scheduling, and communications.

The field of Operations Research emerged in the 1950s as a scientific approach to decision making. Most problems in Operations Research involve a "search for optimality." The max-plus algebra uses the operation of taking a maximum, thus making it an ideal candidate for mathematically describing problems in operations research. Most often, problems in Operations Research have been solved by the development of algorithmic procedures that lead to optimal solutions.

The max-plus algebra emerged in the late 1950s, soon after the field of Operations Research began to develop. This algebraic structure is a semi-ring whose elements are the usual real numbers along with $-\infty$, where the operator of addition, $\bar{\oplus}$, represents taking a maximum and the operator of multiplication, $\otimes$, represents standard addition. Because there is no additive inverse in the max-plus algebra, problem formulation and solutions require different techniques. Although many individuals have researched possible uses and theories regarding max-plus, the first attempt of a complete study, *Minimax Algebra*, by Cuninghame-Green, was not published until 1979 [4]. Many of these initial studies were limited to what are now called path algebras. More recently, the usage of max-plus has been extended to consider Discrete Event Systems and Dynamic Programming [5].

In most of the literature, the basic properties, theorems, and proofs regarding max-plus have become buried in references. Applications of max-plus, although mentioned in the literature, are not usually demonstrated. In this paper, we present the basics properties of max-plus and min-plus, including how to solve systems of max-plus equations, and the properties of max-plus eigenvalues and eigenvectors for irreducible matrices. We then follow up these properties and theorems with applications that motivate the theory, including the shortest route problem, project scheduling, the synchronized event problem, and an airport scheduling problem. In the appendix, there is also a discussion of how the minimum spanning tree problem can be written in terms of min-plus. While max-plus is not necessary to formulate solutions to these applications problems, it is certainly interesting that they can all be formulated and solved using the max-plus (or min-plus) algebra.

Chapter 1: The Max-Plus Algebra

1.1 Basic Properties of Max-plus

The max-plus algebra is an algebraic structure consisting of real numbers where the standard operations of addition and multiplication are replaced by the operation of taking a maximum and the operation of standard addition, respectively. More precisely, let $\mathbb{R}_{\max}$ denote the set $\mathbb{R} \cup \{-\infty\}$, let $\overline{\oplus}$ be a binary operator on $\mathbb{R}_{\max}$ with $x \overline{\oplus} y = \max(x, y)$, and let $\otimes$ be the binary operator on $\mathbb{R}_{\max}$ with $x \otimes y = x + y$. Then the **max-plus algebra** is the algebraic structure consisting of $\mathbb{R}_{\max}$ and the binary operations $\overline{\oplus}$ and $\otimes$.

Interesting outcomes of the use of maximum as the addition operator are the additive identity and the consequent lack of an additive inverse in this system. When we seek an additive identity, we look for an element *z* such that $z \overline{\oplus} x = x$ for all $x \in \mathbb{R}_{\max}$. The only way to guarantee this is to choose $z = -\infty$. Thus, the max-plus algebra has $-\infty$ as its additive identity. Clearly, the operation of taking a maximum is associative and commutative, hence $(\mathbb{R}_{\max}, \overline{\oplus})$ is an abelian semi-group. $(\mathbb{R}_{\max}, \overline{\oplus})$ is not a group, because $x \in \mathbb{R}_{\max}$ has an additive inverse if and only if $x = -\infty$.

A consequence of the lack of additive inverses is that for $a, b \in \mathbb{R},$ the equation $a \overline{\oplus} x = b$ need not have a unique solution. Indeed, the solution to $a \overline{\oplus} x = b$ is $x = b$ if and only if $a < b$. If $a = b$, then the solution for *x* can be any number less than or equal to *b*, and if $a > b$, then $a \overline{\oplus} x = b$ has no solution. The system $a \overline{\oplus} x = -\infty$ has a solution $(x = -\infty)$ only if $a = -\infty$. Since $a \overline{\oplus} a = a$, every element of $\mathbb{R}_{\max}$ is idempotent with respect to $\overline{\oplus}$.

Because $(\mathbb{R}_{\max}, \overline{\oplus})$ is not an abelian group, $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$ does not satisfy the properties of a ring. However, we now show that $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$ satisfies the properties of a commutative semi-ring [5].

Theorem 1.1.1: $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$ satisfies the following properties:

1. $(\mathbb{R}_{\max}, \overline{\oplus})$ is an abelian semi-group.
2. Multiplication is associative and commutative.
3. There is a multiplicative identity.
4. Distributive properties of $\otimes$ over $\overline{\oplus}$, i.e. for $x, y, z \in \mathbb{R}_{\max}$
   a) $z \otimes (x \overline{\oplus} y) = z \otimes x \ \overline{\oplus} \ z \otimes y$
   b) $(x \overline{\oplus} y) \otimes z = x \otimes z \ \overline{\oplus} \ y \otimes z$
5. The additive identity, $-\infty$, is absorbing under multiplication, i.e. for $x \in \mathbb{R}_{\max}$, $-\infty \otimes x = -\infty = x \otimes (-\infty)$.

Proof:

1. This was in the discussion immediately preceding the statement of the theorem.
2. Let $x, y, z \in \mathbb{R}_{\max}$. Then, $x \otimes (y \otimes z) = x + (y + z) = (x + y) + z = (x \otimes y) \otimes z$ and $x \otimes y = x + y = y + x = y \otimes x$.
3. Note $x \otimes 0 = x + 0 = x = 0 + x = 0 \otimes x$. Thus 0 is the multiplicative identity.
4. Let $x, y, z \in \mathbb{R}_{\max}$. Then $z \otimes (x \overline{\oplus} y) = z + \max(x, y) = \max(z + x, z + y)$ $= z \otimes x \ \overline{\oplus} \ z \otimes y$. Statement (b) follows from (a) and the commutativity of $\overline{\oplus}$ and $\otimes$.
5. Let $x \in \mathbb{R}_{\max}$, then $x \otimes (-\infty) = x + \left(-\infty\right) = -\infty$. ■

The max-plus algebra can be extended to matrices. Max-plus matrix addition of matrices is only defined for matrices of the same dimensions. We define the max-plus matrix sum $A \overline{\oplus} B$ to be the matrix resulting from taking entrywise maximums. The max-plus multiplication of a matrix by a scalar results in a matrix where each entry has been increased by the scalar quantity.

Max-plus Matrix Operations:

Let $A = \left[a_{ij}\right]$ and $B = \left[b_{ij}\right]$ be $m \times n$ matrices with entries in $\mathbb{R}_{\max}$ and $c \in \mathbb{R}_{\max}$.

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

$$c \otimes A = \left[c \otimes a_{ij}\right] = \left[c + a_{ij}\right] = \left[a_{ij} + c\right] = A \otimes c$$

Let $A = \left[a_{ij}\right]$ be $m \times n$ and $B = \left[b_{jk}\right]$ be $n \times p$ with entries in $\mathbb{R}_{\max}$.

Then AB is the $m \times p$ matrix whose i, j entry is

$$\left(a_{i1} \otimes b_{1j}\right) \overline{\oplus} \left(a_{i2} \otimes b_{2j}\right) \overline{\oplus} \cdots \overline{\oplus} \left(a_{in} \otimes b_{nj}\right) = \max_{j}\left(a_{ik} + b_{kj}\right).$$

Throughout the paper, we use AB for max-plus multiplication of matrices A and B. As we never use "usual matrix multiplication," this should not lead to difficulties. Because calculations in the max-plus algebra can take some getting used to, we find it is helpful to pause at this point and elaborate with a numerical example.

Example: Matrix Operations

Let $A = \begin{bmatrix} 10 & -\infty \\ 5 & 3 \end{bmatrix}$ and $B = \begin{bmatrix} 8 & 2 \\ 7 & 0 \end{bmatrix}$.

Matrix Addition:

$$A \overline{\oplus} B = \begin{bmatrix} 10 \overline{\oplus} 8 & -\infty \overline{\oplus} 2 \\ 5 \overline{\oplus} 7 & 3 \overline{\oplus} 0 \end{bmatrix} = \begin{bmatrix} 10 & 2 \\ 7 & 3 \end{bmatrix}$$

Scalar Multiplication:

$$5 \otimes A = \begin{bmatrix} 5 \otimes 10 & 5 \otimes (-\infty) \\ 5 \otimes 5 & 5 \otimes 3 \end{bmatrix} = \begin{bmatrix} 15 & -\infty \\ 10 & 8 \end{bmatrix}$$

Matrix Multiplication:

$$\begin{aligned} AB &= \begin{bmatrix} 10 & -\infty \\ 5 & 3 \end{bmatrix} \begin{bmatrix} 8 & 2 \\ 7 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 10 \otimes 8 \ \overline{\oplus} \ (-\infty) \otimes 7 & 10 \otimes 2 \ \overline{\oplus} \ (-\infty) \otimes 0 \\ 5 \otimes 8 \ \overline{\oplus} \ 3 \otimes 7 & 5 \otimes 2 \ \overline{\oplus} \ 3 \otimes 0 \end{bmatrix} \\ &= \begin{bmatrix} 18 \overline{\oplus} (-\infty) & 12 \overline{\oplus} (-\infty) \\ 13 \overline{\oplus} 10 & 7 \overline{\oplus} 3 \end{bmatrix} = \begin{bmatrix} 18 & 12 \\ 13 & 7 \end{bmatrix} \end{aligned}$$

Theorem 1.1.2: Matrix multiplication in $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$ is associative, but not necessarily commutative.

Proof:

Let A be $m \times n$, B be $n \times p$, and C be $p \times q$ matrices with entries from $\mathbb{R}_{\max}$.

The i, ℓ entry of $(AB)C$ is $\max\limits_k \left(\left(\max\limits_j \left(a_{ij} + b_{jk} \right) \right) + c_{k\ell} \right) = \max\limits_{k,j} \left(a_{ij} + b_{jk} + c_{k\ell} \right)$.

The i, ℓ entry of $A(BC)$ is $\max\limits_j \left(a_{ij} + \left(\max\limits_k \left(b_{jk} + c_{k\ell} \right) \right) \right) = \max\limits_{j,k} \left(a_{ij} + b_{jk} + c_{k\ell} \right)$.

Thus $(AB)C = A(BC)$, and max-plus matrix multiplication is associative.

To show that matrix multiplication is not necessarily commutative, consider the following simple counterexample:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 4 & 4 \end{pmatrix} \text{ is not equal to } \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 3 & 4 \end{pmatrix}. \quad \blacksquare$$

Powers in the max-plus algebra correspond to the usual scalar multiplication in the real numbers. Let n be a positive integer and $a \in \mathbb{R}$. Then we define a^n as follows:

$$a^n = \underbrace{a \otimes a \otimes \cdots \otimes a}_{n \text{ times}} = \underbrace{a + a + \cdots + a}_{n \text{ times}} = n \cdot a$$

For the max-plus algebra, we also include $-\infty$ in the domain. Thus, we need only verify that the rule $a^n = n \cdot a$ will hold for $a = -\infty$ and $n > 0$.

$$(-\infty)^n = \underbrace{(-\infty) \otimes (-\infty) \otimes \cdots \otimes (-\infty)}_{n \text{ times}} = \underbrace{(-\infty) + (-\infty) + \cdots + (-\infty)}_{n \text{ times}} = n \cdot (-\infty) = -\infty$$

Theorem 1.1.3: In $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$, with $x, y \in \mathbb{Z}^+$ and $a \in \mathbb{R}_{\max}$, the following exponent properties hold:

1. $a^x \otimes a^y = a^{x+y}$
2. $(a^x)^y = a^{x \cdot y}$

Proof:

Let $a \in \mathbb{R}$ and $x, y \in \mathbb{Z}^+$.

1. $a^x \otimes a^y = x \cdot a + y \cdot a = (x + y) \cdot a = a^{x+y}$

   $(-\infty)^x \otimes (-\infty)^y = (-\infty) \otimes (-\infty) = (-\infty) + (-\infty) = -\infty = (-\infty)^{x+y}$
2. $(a^x)^y = (x \cdot a)^y = y \cdot (x \cdot a) = (x \cdot y) \cdot a = a^{x \cdot y}$

   $\left((-\infty)^x\right)^y = (-\infty)^y = -\infty = (-\infty)^{x \cdot y} \quad \blacksquare$

Notice that for $x \in \mathbb{R}$, we can find y such that $x \otimes y = 0$ (recall that 0 is the multiplicative identity in max-plus). So each element $x \in \mathbb{R}$ has a multiplicative inverse $x^{-1} = -x$. In $(\mathbb{R}, +, \cdot)$, there is no multiplicative inverse for zero, and similarly, in $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$, there is no multiplicative inverse for the additive identity, $-\infty$. From this definition of the multiplicative inverse, we can extend the properties in Theorem 1.1.3 to include $x, y \in \mathbb{Z}$ with the footnote that if $n > 0$, then $(-\infty)^{-n}$ will be undefined.

In optimization problems where one is searching for some kind of minimization, it is easier to formulate the problems in terms of a related algebraic structure, called the **min-plus algebra**. The min-plus algebraic structure has elements $\mathbb{R}_{\min} = \mathbb{R} \bigcup \{\infty\}$ with binary operations $\underline{\oplus}$ and $\otimes$. For $x, y \in \mathbb{R}_{\min}$, $x \underline{\oplus} y = \min(x, y)$ and $x \otimes y = x + y$. Indeed, $(\mathbb{R}_{\min}, \underline{\oplus}, \otimes)$ and $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$ are isomorphic algebraic structures. Let $\theta : \mathbb{R}_{\max} \to \mathbb{R}_{\min}$ by $\theta(x) = -x$. Then $\theta(x \overline{\oplus} y) = -\max(x, y) = \min(-x, -y) = \theta(x) \underline{\oplus} \theta(y)$ and $\theta(x \otimes y) = -(x + y) = (-x) + (-y) = \theta(x) \otimes \theta(y)$. Thus, the result about the max-plus algebra can be translated into a result about the min-plus algebra, that is, $(\mathbb{R}_{\min}, \underline{\oplus}, \otimes)$ is also a commutative semi-ring. In particular, we will see the min-plus algebra used in the shortest route problem (Section 2.1). Commonly in papers, the operation $\oplus$ can be used to denote either a maximum or minimum, depending on the situation. To distinguish between these cases and avoid confusion, in this paper we have adopted the notation $\overline{\oplus}$ for maximum and $\underline{\oplus}$ for minimum.

1.2 Solving Systems of Equations in Max-plus

In this section, we develop the theory of linear systems of equations for the max-plus arithmetic [3, 4]. Although there are some parallels between solving systems of equations in $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$ and in $(\mathbb{R}, +, \cdot)$, the operation $\overline{\oplus}$ creates some interesting differences. In general, we would like to be able to solve the matrix equation $A\mathbf{x} = \mathbf{b}$, where A is an $m \times n$ matrix, $\mathbf{x}$ is an $n \times 1$ vector, and $\mathbf{b}$ is an $m \times 1$ vector. It will be helpful if we look at the equivalent system of equations in the usual arithmetic to first get an idea for how to go about solving the system. We can rewrite $A\mathbf{x} = \mathbf{b}$ as the following detailed matrix equation and then the equivalent system of Max-plus equations:

$$A\mathbf{x} = \mathbf{b} \qquad \Leftrightarrow \qquad \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

$$\Leftrightarrow \qquad \begin{array}{c} (a_{11} \otimes x_1) \overline{\oplus} (a_{12} \otimes x_2) \overline{\oplus} \cdots \overline{\oplus} (a_{1n} \otimes x_n) = b_1 \\ (a_{21} \otimes x_1) \overline{\oplus} (a_{22} \otimes x_2) \overline{\oplus} \cdots \overline{\oplus} (a_{2n} \otimes x_n) = b_2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ (a_{m1} \otimes x_1) \overline{\oplus} (a_{m2} \otimes x_2) \overline{\oplus} \cdots \overline{\oplus} (a_{mn} \otimes x_n) = b_m \end{array}$$

Written in standard notation, we must simultaneously solve the following system:

$$\begin{array}{c} \max\{(a_{11} + x_1), (a_{12} + x_2), \ldots, (a_{1n} + x_n)\} = b_1 \\ \max\{(a_{21} + x_1), (a_{22} + x_2), \ldots, (a_{2n} + x_n)\} = b_2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ \max\{(a_{m1} + x_1), (a_{m2} + x_2), \ldots, (a_{mn} + x_n)\} = b_m \end{array}$$

We first consider the case that a solution exists and some of the entries of $\mathbf{b}$ are $-\infty$. Without loss of generality, we can reorder the equations so that the finite entries of $\mathbf{b}$ occur first:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_k \\ \hline -\infty \\ \vdots \\ -\infty \end{pmatrix}$$

Written in standard notation, this gives the following system of equations:

$$\begin{cases} \max\left(a_{11}+x_1,\ a_{12}+x_2,\ \dots\ ,\ a_{1n}+x_n\right)=b_1 \\ \ \vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ \max\left(a_{k1}+x_1,\ a_{k2}+x_2,\ \dots\ ,\ a_{kn}+x_n\right)=b_k \\ \max\left(a_{k+1,1}+x_1,\ a_{k+1,2}+x_2,\ \dots\ ,\ a_{k+1,n}+x_n\right)=-\infty \\ \ \vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ \max\left(a_{n1}+x_1,\ a_{n2}+x_2,\ \dots\ ,\ a_{nn}+x_n\right)=-\infty \end{cases}$$

We can renumber the variables so that those j such that $a_{k+1,j},\dots,a_{m,j}=-\infty$ occur first:

$$\left(\begin{array}{ccc|c} & A_1 & & A_2 \\ \hline -\infty & \cdots & -\infty & \\ \vdots & \ddots & \vdots & A_3 \\ -\infty & \cdots & -\infty & \end{array}\right)\begin{pmatrix} x_1 \\ \vdots \\ x_\ell \\ \hline x_{\ell+1} \\ \vdots \\ x_n \end{pmatrix}=\begin{pmatrix} b_1 \\ \vdots \\ b_k \\ \hline -\infty \\ \vdots \\ -\infty \end{pmatrix}$$

Let the dimensions of A_1 be $k\times\ell$. Let $\mathbf{b}'=\begin{pmatrix} b_1 \\ \vdots \\ b_k \end{pmatrix}$ and $\mathbf{x}'=\begin{pmatrix} x_1 \\ \vdots \\ x_\ell \end{pmatrix}$. Note that if $A\mathbf{x}=\mathbf{b}$ has a solution, then $x_{k+1}=x_n=-\infty$, and $A\mathbf{x}'=\mathbf{b}'$. Thus, $A\mathbf{x}=\mathbf{b}$ has a solution if and only if $\mathbf{x}'$ is a solution to $A_1\mathbf{x}'=\mathbf{b}'$ and solutions to $A\mathbf{x}=\mathbf{b}$ are

$$\mathbf{x}=\begin{pmatrix} \mathbf{x}' \\ -\infty \\ \vdots \\ -\infty \end{pmatrix}.$$

Therefore, the solvability of a system with infinite entries in $\mathbf{b}$ can be reduced to that of a system where all the entries in $\mathbf{b}'$ are finite. Hence, we restrict our attention to systems $A\mathbf{x}=\mathbf{b}$ where all the entries of $\mathbf{b}$ are finite. If there is to be a solution to the system of max-plus equations, then $a_{ij}+x_j\le b_i$ for all $i\in\{1,\dots,m\}$ and $j\in\{1,\dots,n\}$. To search for a solution to the system, we first consider each component of $\mathbf{x}$ separately. Consider, for example, x_1. If there is a solution to the system, then $a_{i1}+x_1\le b_i$ for $i=1,2,\dots,m$. Thus $x_1\le b_i-a_{i1}$ for each i, leading us to the following system of upper bounds on x_1:

$$\begin{aligned} x_1 &\le b_1 - a_{11} \\ x_1 &\le b_2 - a_{21} \\ &\vdots \\ x_1 &\le b_m - a_{m1} \end{aligned}$$

If this system of inequalities has a solution, then it satisfies

$$x_1 \le \min\{(b_1 - a_{11}), (b_2 - a_{21}), \ldots, (b_m - a_{m1})\}.$$

Similarly, we can find the possible solutions for $x_2, \ldots, x_n$, giving us the following system of inequalities on the entries of $\mathbf{x}$:

$$\begin{array}{l} x_1 \le \min\{(b_1 - a_{11}), (b_2 - a_{21}), \ldots, (b_m - a_{m1})\} \\ x_2 \le \min\{(b_1 - a_{12}), (b_2 - a_{22}), \ldots, (b_m - a_{m2})\} \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad \vdots \qquad \vdots \\ x_n \le \min\{(b_1 - a_{1n}), (b_2 - a_{2n}), \ldots, (b_m - a_{mn})\} \end{array}$$

This leads us to a candidate for a solution to $A\mathbf{x} = \mathbf{b}$, which we will denote by $\mathbf{x}'$.

$$\mathbf{x}' = \begin{pmatrix} x_1' \\ x_2' \\ \vdots \\ x_n' \end{pmatrix} \quad \text{where} \quad \begin{cases} x_1' = \min\{(b_1 - a_{11}), (b_2 - a_{21}), \ldots, (b_m - a_{m1})\} \\ x_2' = \min\{(b_1 - a_{12}), (b_2 - a_{22}), \ldots, (b_m - a_{m2})\} \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad \vdots \qquad \vdots \\ x_n' = \min\{(b_1 - a_{1n}), (b_2 - a_{2n}), \ldots, (b_m - a_{mn})\} \end{cases}$$

Before we pause for a numerical example, let us introduce another matrix to simplify the process of solving a system of max-plus equations. We define the discrepancy matrix, $D_{A,\mathbf{b}}$ as follows:

$$D_{A,\mathbf{b}} = \begin{pmatrix} b_1 - a_{11} & b_1 - a_{12} & \cdots & b_1 - a_{1n} \\ b_2 - a_{21} & b_2 - a_{22} & \cdots & b_2 - a_{2n} \\ \vdots & & \ddots & \vdots \\ b_m - a_{m1} & b_m - a_{m2} & \cdots & b_m - a_{mn} \end{pmatrix}$$

Note that $D_{A,\mathbf{b}}$ is simply a matrix with all the upper bounds of the x_i's and that each x_i' can be found by taking the minimum of the ith column of $D_{A,\mathbf{b}}$.

Example 1.2a: Max-Plus System with only one solution

Solve $A\mathbf{x} = \mathbf{b}$ where $A = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 4 & 6 \\ 3 & 1 & -2 \\ 9 & 6 & 3 \end{pmatrix}$, $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, and $\mathbf{b} = \begin{pmatrix} 6 \\ 10 \\ 5 \\ 11 \end{pmatrix}$.

A quick calculation gives the discrepancy matrix: $D_{A,\mathbf{b}} = \begin{pmatrix} 4 & 3 & 5 \\ 10 & 6 & 4 \\ 2 & 4 & 7 \\ 2 & 5 & 8 \end{pmatrix}$

The entries for the candidate solution can be found by taking the minimum of each column of $D_{A,\mathbf{b}}$.

$$x_1' = \min(4,10,2,2) = 2$$
$$x_2' = \min(3,6,4,5) = 3$$
$$x_3' = \min(5,4,7,8) = 4$$

Thus, $\mathbf{x}' = (2 \quad 3 \quad 4)^T$ is the candidate solution to $A\mathbf{x} = \mathbf{b}$. We can verify that this is indeed a solution to $A\mathbf{x} = \mathbf{b}$ by plugging it back in:

$$\begin{pmatrix} 2 & 3 & 1 \\ 0 & 4 & 6 \\ 3 & 1 & -2 \\ 9 & 6 & 3 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \max(4,6,5) \\ \max(2,7,10) \\ \max(5,4,2) \\ \max(11,9,7) \end{pmatrix} = \begin{pmatrix} 6 \\ 10 \\ 5 \\ 11 \end{pmatrix}$$

As we shall see, this will be the only solution to the matrix equation. ■

Example 1.2b: Max-Plus System with no solution

Solve $A\mathbf{x} = \mathbf{b}$ where $A = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 4 & 6 \\ 3 & 1 & -2 \\ 9 & 6 & 3 \end{pmatrix}$, $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, and $\mathbf{b} = \begin{pmatrix} 6 \\ 12 \\ 5 \\ 9 \end{pmatrix}$.

Then $D_{A,\mathbf{b}} = \begin{pmatrix} 4 & 3 & 5 \\ 12 & 8 & 6 \\ 2 & 4 & 7 \\ 0 & 3 & 6 \end{pmatrix}$, which gives us the candidate solution of $\mathbf{x}' = \begin{pmatrix} 0 & 3 & 5 \end{pmatrix}^T$.

When we try $\mathbf{x}'$ in the matrix equation, we see that it is not a solution:

$$\begin{pmatrix} 2 & 3 & 1 \\ 0 & 4 & 6 \\ 3 & 1 & -2 \\ 9 & 6 & 3 \end{pmatrix} \begin{pmatrix} 0 \\ 3 \\ 5 \end{pmatrix} = \begin{pmatrix} \max(2,6,6) \\ \max(0,7,11) \\ \max(3,4,3) \\ \max(9,9,8) \end{pmatrix} = \begin{pmatrix} 6 \\ \mathbf{11} \\ \mathbf{4} \\ 9 \end{pmatrix} \neq \quad \mathbf{b} = \begin{pmatrix} 6 \\ 12 \\ 5 \\ 9 \end{pmatrix}$$

The bolded entries do not match the corresponding entries of $\mathbf{b}$. Since the components of $\mathbf{x}'$ are the upper bounds, we know that a solution $\mathbf{x}$ must satisfy $x_1 \le 0$, $x_2 \le 3$, and $x_3 \le 5$. But then, from the second row, we have $\max(x_1 + 0, x_2 + 4, x_3 + 6) \le 11 < 12$. Thus, this matrix equation has no solution. ■

Example 1.2c: Max-Plus System with infinitely many solutions

Solve $A\,\mathbf{x} = \mathbf{b}$ where $A = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 4 & 6 \\ 3 & 1 & -2 \\ 9 & 6 & 3 \end{pmatrix}$, $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, and $\mathbf{b} = \begin{pmatrix} 8 \\ 13 \\ 5 \\ 10 \end{pmatrix}$.

Then $D_{A,\mathbf{b}} = \begin{pmatrix} 6 & 5 & 7 \\ 13 & 9 & 7 \\ 2 & 4 & 7 \\ 1 & 4 & 7 \end{pmatrix}$, which gives us the candidate solution of $\mathbf{x}' = \begin{pmatrix} 1 & 4 & 7 \end{pmatrix}^T$.

Check $\mathbf{x}'$ is a solution: $\begin{pmatrix} 2 & 3 & 1 \\ 0 & 4 & 6 \\ 3 & 1 & -2 \\ 9 & 6 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix} = \begin{pmatrix} \max(3,7,8) \\ \max(1,8,13) \\ \max(4,5,5) \\ \max(10,10,10) \end{pmatrix} = \begin{pmatrix} 8 \\ 13 \\ 5 \\ 10 \end{pmatrix}$

Thus $\mathbf{x}'$ is a solution to the given matrix equation. But notice that there are other solutions that also work. Indeed, each $\mathbf{x}$ of the form $\{\mathbf{x} : \mathbf{x} = (a, b, 7)^T$ where $a \le 1$ and $b \le 4\}$ is also a solution to $A\,\mathbf{x} = \mathbf{b}$. ■

In order to predict the number of solutions to the matrix equation $A\,\mathbf{x} = \mathbf{b}$, it will be useful to define a "reduced" discrepancy matrix, $R_{A,\mathbf{b}}$:

$$R_{A,\mathbf{b}} = (r_{ij}) \text{ where } r_{ij} = \begin{cases} 1 & \text{if } d_{ij} = \text{minimum of column } j \\ 0 & \text{otherwise} \end{cases}$$

Below we show $D_{A,\mathbf{b}}$ and $R_{A,\mathbf{b}}$ for the examples 1.2a, b, and c. The bold entries for each $D_{A,\mathbf{b}}$ show where the minimum occurs in each column. Notice that these are the 'one' entries of each corresponding $R_{A,\mathbf{b}}$.

Example 1.2a: One solution	**Example 1.2b:** No solutions	**Example 1.2c:** Infinite solutions
$D_{A,\mathbf{b}} = \begin{pmatrix} 4 & \mathbf{3} & 5 \\ 10 & 6 & \mathbf{4} \\ \mathbf{2} & 4 & 7 \\ \mathbf{2} & 5 & 8 \end{pmatrix}$	$D_{A,\mathbf{b}} = \begin{pmatrix} 4 & \mathbf{3} & \mathbf{5} \\ 12 & 8 & 6 \\ 2 & 4 & 7 \\ \mathbf{0} & \mathbf{3} & 6 \end{pmatrix}$	$D_{A,\mathbf{b}} = \begin{pmatrix} 6 & 5 & \mathbf{7} \\ 13 & 9 & \mathbf{7} \\ 2 & \mathbf{4} & \mathbf{7} \\ \mathbf{1} & \mathbf{4} & \mathbf{7} \end{pmatrix}$
$R_{A,\mathbf{b}} = \begin{pmatrix} 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & 0 \end{pmatrix}$	$R_{A,\mathbf{b}} = \begin{pmatrix} 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 \end{pmatrix}$	$R_{A,\mathbf{b}} = \begin{pmatrix} 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} \\ 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix}$

Recall that for a column, j, in the $D_{A,\mathbf{b}}$ matrix, the minimum entry of the column is the maximum solution to the system of inequalities for x_j. In order to change this system of inequalities to a system of equalities, we must have equality in each row inequality, i.e. there must be at least one minimum in each row of $D_{A,\mathbf{b}}$, i.e. there must be at least one 1 in each row of $R_{A,\mathbf{b}}$ for a solution to exist. Indeed, we see that in example 1.2b, there are rows of $R_{A,\mathbf{b}}$ that contain no 1's (zero-rows).

Theorem 1.2.1: Let $A\,\mathbf{x} = \mathbf{b}$ be a matrix equation in $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$ where A is an $m \times n$ matrix, and $\mathbf{b}$ is a $n \times 1$ vector with all entries finite.

a) If there is a zero-row in the reduced discrepancy matrix, $R_{A,\mathbf{b}}$, then there is no solution to the matrix equation.

b) If there is at least one 1 in each row of the reduced discrepancy matrix, $R_{A,\mathbf{b}}$, then $\mathbf{x}'$ is a solution to $A\,\mathbf{x} = \mathbf{b}$.

Proof:

a) Without loss of generality, denote the zero-row of $R_{A,\mathbf{b}}$ by row *k*. Suppose to the contrary that $\tilde{\mathbf{x}}$ is a solution of $A\,\mathbf{x} = \mathbf{b}$. Then $\tilde{x}_j \le \min\limits_{\ell}\left(b_\ell - a_{\ell j}\right) < b_k - a_{kj}$. Thus

$\tilde{x}_j + a_{kj} < b_k$ for all *j*. Hence, $\tilde{\mathbf{x}}$ does not satisfy the *k*th equation and is not a solution to $A\,\mathbf{x} = \mathbf{b}$.

b) We prove the contrapositive. Suppose $\mathbf{x}'$ is not a solution to the matrix equation. By definition, $x'_j \le b_k - a_{kj}$ for all j,k. Hence $\max_j(a_{kj} + x'_j) \le b_{kj}$ and if $\mathbf{x}'$ is not a solution then there is a *k* with $\max_j(a_{kj} + x'_j) < b_k$. This is equivalent to $x'_j < b_k - a_{kj}$ for all *j*. Since $x'_j = \min\left(b_\ell - a_{\ell j}\right)$ for some ℓ, there is no entry in row *k* of $R_{A,\mathbf{b}}$ that is 1. ■

Now, provided we know that a solution to $A\,\mathbf{x} = \mathbf{b}$ exists, how can we tell the number of solutions to this equation? We need to define the concept of fixed entries in $R_{A,\mathbf{b}}$.

Definition: The 1 in a row of $R_{A,\mathbf{b}}$ is a **variable-fixing entry** if either

a) it is the only 1 in that row (a **lone-one**), or

b) it is in the same column as a lone-one.

The remaining 1s are called **slack entries**.

A 1 in the *j*th column of $R_{A,\mathbf{b}}$ signifies the minimum of the upper bounds for x_j. If there are no other ones in the row where a one occurs, then the only way that the equation corresponding to that row can be solved is to have x_j achieve the bound. This causes the value of x_j to be fixed at a specific value, i.e. it is a variable-fixing entry. To illustrate this principle, we circle the variable-fixing entries for the previous examples in the following table.

Example 1.2a: One solution	**Example 1.2b:** No solutions	**Example 1.2c:** Infinite solutions
$R_{A,\mathbf{b}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$	$R_{A,\mathbf{b}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$	$R_{A,\mathbf{b}} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

In Example 1.2a, all of the non-zero entries are variable-fixing entries. The first row equation fixes the x_2 component, $x_2 = 3$. The second row equation fixes the x_3 component, $x_3 = 4$. The third row equation fixes the x_1 component, $x_1 = 2$. When we reach the fourth row equation, all the components of $\mathbf{x}$ have been chosen. None of the components can be changed without causing an inequality in one of the first three rows.

In Example 1.2c, there are slack entries in $R_{A,\mathbf{b}}$. The first row equation fixes the x_3 component, $x_3 = 7$. The component solution to the second row equation has already been fixed by the first row equation. In the third row equation, there are two possible ways to achieve equality – with either $x_2 = 4$ or $x_3 = 7$. But we have already fixed the x_3 component to be 7. So now as long as $x_2 \le 4$, we will not cause any problems in this or any of the row equations above it. Similarly, in the fourth row equation, the equality can be satisfied by using the already fixed component of $x_3 = 7$. As long as $x_1 \le 1$ and $x_2 \le 4$, the row equation will always be true.

The following theorem shows that in order for $A\mathbf{x} = \mathbf{b}$ to have a unique solution, each component of $\mathbf{x}$ must be fixed, i.e. there can be no slack entries (a slack entry can only exist if there are no variable-fixing entries in that column). Thus, for $A\mathbf{x} = \mathbf{b}$ to have a unique solution, there must be a lone-one in each column.

Theorem 1.2.2: Let $A\mathbf{x} = \mathbf{b}$ be a matrix equation in $(\mathbb{R}_{\max}, \overline{\oplus}, \otimes)$ where A is an $m \times n$ matrix, $\mathbf{b}$ is an $n \times 1$ vector with finite entries, and a solution to the equation exists.

a) If each column of $R_{A,\mathbf{b}}$ has a lone-one, then the solution to the matrix equation is unique.
b) If there are slack entries in $R_{A,\mathbf{b}}$, then there are infinite solutions to the matrix equation.

Proof:

a) If there is a lone-one in each column of $R_{A,\mathbf{b}}$, then there is a variable-fixing entry in each column of $R_{A,\mathbf{b}}$. There can be no slack entries since all the columns contain a variable-fixing entry. All the components of $\mathbf{x}$ are fixed and thus the solution is unique.
b) Let r_{ij} be one of the slack entries in $R_{A,\mathbf{b}}$ and let $\tilde{\mathbf{x}}$ be a solution to the equation $A\mathbf{x} = \mathbf{b}$. Since r_{ij} is not fixed, then there are no fixed entries in the *j*th column of $R_{A,\mathbf{b}}$. Thus, equality can be achieved for each row equation without using the $\tilde{x}_j$

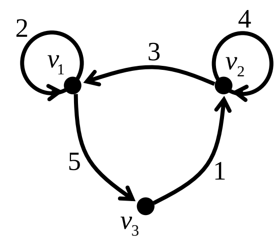
component. Thus, while the value of $\tilde{x}_j$ does indicate the maximum value possible for this component, any smaller value will not alter the existence of equalities in the row equations. ■

It is interesting to note that in $(\mathbb{R},+,\cdot)$, an $m \times n$ system of linear equations has either no solution, one solution, or an infinite number of solutions. Similarly, for an $m \times n$ system of linear equations in $(\mathbb{R}_{\max},\overline{\oplus},\otimes)$, we also have either no solution, one solution, or an infinite number of solutions.

1.3 Max-plus Eigenvalues and Eigenvectors

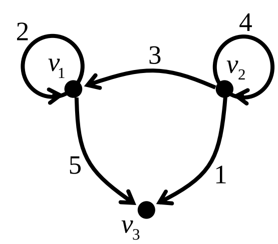
Before we look at the eigenproblem in the max-plus setting, we review some terminology from graph theory [1, 2]. For an $n \times n$ matrix, *A*, we can define the **digraph** (or directed graph) of *A*, as the graph with vertices $1, \dots, n$ where there is a directed arc from i to j with weight a_{ij} if and only if $a_{ij} \neq -\infty$. A **path** is a sequence of distinct vertices $i_1, i_2, \dots, i_k$ such that there is an arc from i_j to i_{j+1} for $j = 1, \dots, k-1$. We can refer to the weight of a path as the sum of the weights of the arcs that make up that path. The digraph, D_A, is **strongly connected** if there is a path from any vertex to any other vertex. If D_A is strongly connected, then we say the matrix *A* is **irreducible**. For example, the matrix *A* below is irreducible:

$$A = \begin{pmatrix} 2 & -\infty & 5 \\ 3 & 4 & -\infty \\ -\infty & 1 & -\infty \end{pmatrix} \quad \text{gives the digraph}$$



The matrix B (shown below) is reducible since there is no path from v_3 to any other vertex.

$$B = \begin{pmatrix} 2 & -\infty & 5 \\ 3 & 4 & 1 \\ -\infty & -\infty & -\infty \end{pmatrix} \quad \text{gives the digraph}$$



A **cycle,** σ**,** is a sequence, $i_1, i_2, \dots, i_k$, of distinct vertices such that $i_1 \to i_2$, $i_2 \to i_3, \dots, i_k \to i_1$. of adjacent arcs in the digraph that starts and ends at the same vertex and does not travel through any other vertex more than once. When we discuss the cycle in reference to a digraph, it can be described as a sequence of vertices, $\sigma : i \to j \to \cdots * \to i$. We can also refer to a cycle within a matrix, $\sigma : a_{ij}, a_{jk}, \dots, a_{*i}$. The number of arcs in a cycle is called the **length**, ℓ_σ. Note that for any σ, $\ell_\sigma \leq n$. In a strongly connected graph, there must be at least one cycle originating at each vertex. For matrix *A* given in the first example, $v_1 \to v_2 \to v_3 \to v_1$ has length 3 and originates at v_1. A **loop** is a cycle with length 1; in the digraph D_A, shown above, there are loops at v_1 and v_2. For a cycle σ, the sum of its arc weights divided by the length, ℓ_σ is called the **mean**, $M(\sigma)$. For a matrix *A* with distinct cycles $\sigma_1, \sigma_2, \dots, \sigma_n$, we define the

maximum cycle mean by $\mu(A) = \max_i M(\sigma_i)$. A graph that contains only the cycles with the maximum cycle mean is called a **critical graph**.

Let A be an $n \times n$ matrix with entries from $\mathbb{R}_{\max}$. Then we define $\lambda \in \mathbb{R}$ to be the **eigenvalue** of *A* with **eigenvector** $\mathbf{x}$, where at least one entry is not $-\infty$, provided λ and $\mathbf{x}$ satisfy the max-plus equation $A\,\mathbf{x} = \lambda \otimes \mathbf{x}$. We refer to $(\lambda, \mathbf{x})$ as an **eigenpair** for *A*. When we find a particular eigenvector, any max-plus scalar multiple of it is also an eigenvector (Lemma 1.3.1). When we refer to a unique eigenvector, we include the scalar max-plus multiples in the uniqueness.

Lemma 1.3.1: Let *A* be an $n \times n$ matrix with eigenpair $(\lambda, \mathbf{x})$ and $c \in \mathbb{R}$. Then $(\lambda, c \otimes \mathbf{x})$ is also an eigenpair of *A*.

Proof:

Suppose $(\lambda, \mathbf{x})$ is an eigenpair for the irreducible $n \times n$ matrix *A*. Then we have $A\mathbf{x} = \lambda \otimes \mathbf{x}$. Multiplying both sides of the equation by the scalar *c*, and then using the commutativity of max-plus scalar multiplication, and the associativity of max-plus matrix multiplication:

$$\begin{aligned} c \otimes A\mathbf{x} &= c \otimes \lambda \otimes \mathbf{x} \\ (A \otimes c)\mathbf{x} &= \lambda \otimes c \otimes \mathbf{x} \\ A(c \otimes \mathbf{x}) &= \lambda \otimes (c \otimes \mathbf{x}) \quad \blacksquare \end{aligned}$$

There are many known results for the topic of max-plus eigenvalues and eigenvectors [1, 3, 4]; in this paper, we present the results that pertain to irreducible matrices, since this case is used in the application section 2.4. Before we consider the irreducible matrix, there are several results that we first prove for the case of a matrix with at least one cycle.

Lemma 1.3.2: Let *A* be an $n \times n$ matrix with at least one cycle, then

a) *A* has finite eigenvalue *k* if and only if $-k \otimes A$ has eigenvalue 0.

b) $\mu(A) = m$ if and only if $\mu(-m \otimes A) = 0$.

Proof:

a) Let $(0, \mathbf{x})$ be the eigenpair for $-k \otimes A$.

Then we have the following equivalent statements:

$$(-k \otimes A)\mathbf{x} = 0 \otimes \mathbf{x} \quad \Leftrightarrow \quad \begin{pmatrix} a_{11} - k & \cdots & a_{1n} - k \\ \vdots & \ddots & \vdots \\ a_{n1} - k & \cdots & a_{nn} - k \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = 0 \otimes \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$$\Leftrightarrow \begin{cases} \max\left(a_{11} - k + x_1,\ a_{12} - k + x_2, \ldots,\ a_{1n} - k + x_n\right) = x_1 \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \max\left(a_{n1} - k + x_1,\ a_{n2} - k + x_2, \ldots,\ a_{nn} - k + x_n\right) = x_n \end{cases}$$

$$\Leftrightarrow \begin{cases} -k \otimes \max\left(a_{11} + x_1,\ a_{12} + x_2, \ldots,\ a_{1n} + x_n\right) = x_1 \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ -k \otimes \max\left(a_{n1} + x_1,\ a_{n2} + x_2, \ldots,\ a_{nn} + x_n\right) = x_n \end{cases}$$

$$\Leftrightarrow \begin{cases} \max\left(a_{11} + x_1,\ a_{12} + x_2, \ldots,\ a_{1n} + x_n\right) = k \otimes x_1 \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \max\left(a_{n1} + x_1,\ a_{n2} + x_2, \ldots,\ a_{nn} + x_n\right) = k \otimes x_n \end{cases}$$

$$\Leftrightarrow \quad A\,\mathbf{x} = k \otimes \mathbf{x}$$

b) Let $\mu(A) = m$ and let $B = -m \otimes A$. Then the weight of any arc in D_B has been decreased by *m* from the corresponding arc in *A*, i.e. $b_{ij} = a_{ij} - m$. Let the cycle $\sigma_A : a_{ij}, a_{jk}, \ldots, a_{*i}$ have the maximum cycle mean in *A* with length ℓ. Then

$$\mu(A) = \frac{a_{ij} + a_{jk} + \cdots + a_{*i}}{\ell} = m.$$

Let $\sigma_B : b_{ij}, b_{jk}, \ldots, b_{*i}$ be the same cycle taken, only taken from *B*, so then

$$\begin{aligned} \mu(B) &= \frac{b_{ij} + b_{jk} + \cdots b_{*i}}{\ell} \\ &= \frac{(a_{ij} - m) + (a_{jk} - m) + \cdots + (a_{*i} - m)}{\ell} \\ &= \frac{a_{ij} + a_{jk} + \cdots a_{*i} - mk}{\ell} \\ &= \mu(A) - m \\ &= 0. \end{aligned}$$

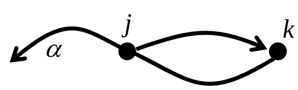
■

Theorem 1.3.3: Let A be an $n \times n$ matrix with at least one cycle. Then there exists an eigenpair $(\lambda, \mathbf{x})$ for A where λ is finite.

Proof:

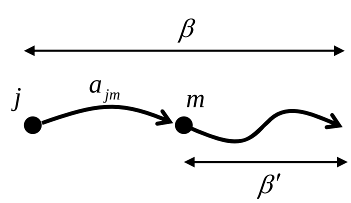
Without loss of generality, by Lemma 1.3.2, we may assume A has a maximum cycle mean of 0. It will be sufficient to show that A has eigenvalue 0. Let D_A be the digraph for A and let $\mathbf{x}$ be the vector whose i th entry is the largest weight of a path in D_A that starts at i. If there is no path in D_A that starts at i, then $x_i = -\infty$. The j th entry of $A\mathbf{x}$ is $\max_k \left(a_{jk} + x_k\right)$.

Consider $a_{jk} + x_k$. Suppose $a_{jk} = -\infty$, then $a_{jk} + x_k = -\infty \le x_j$. Suppose $a_{jk} \ne -\infty$, i.e. there is an arc from j to k. Let α be a path that starts at k with largest weight, x_k.



Let $\alpha' = j \to k, \alpha$. α' may contain a cycle (α could pass through vertex j), but then α' can be decomposed into a path that starts at j and a cycle. Since each cycle has non-positive mean, α' has weight at most x_j. Hence, $a_{jk} + x_k \le x_j$. Therefore, we've shown that $\max_k \left(a_{jk} + x_k\right) \le x_j$.

Now let β be a path in D_A with largest weight, x_j that starts at j. Then β is $j \to m, \beta'$ for some vertex m and some path β' starting at m.



So the weight of β is $x_j = a_{jm} + \mathrm{wt}\left(\beta'\right) \le a_{jm} + x_m \le \max_k \left(a_{jk} + x_k\right)$. Thus, we now have that $\max_k \left(a_{jk} + x_k\right) = x_j$. Therefore, $\mathbf{x}$ is an eigenvector of A corresponding to eigenvalue 0. ■

We now look at the eigenvalue for an irreducible matrix. It is easy to verify that for an irreducible matrix, every vertex is the vertex of a cycle. We have already shown that if a matrix has a cycle, then the eigenvalue is finite. Thus, an irreducible matrix must also have only finite eigenvalues.

Theorem 1.3.4: Let *A* be an irreducible $n \times n$ matrix. Then $\lambda = \mu(A)$ is the unique finite eigenvalue of *A*.

Proof:

Let *A* be an irreducible $n \times n$ matrix with entries in $\mathbb{R}_{\max}$ and D_A be the digraph for *A*. Let $(\lambda, \mathbf{x})$ be an eigenpair of *A* with λ finite, then this eigenpair satisfies the equation $A\mathbf{x} = \lambda \otimes \mathbf{x}$. We first argue that all the entries of $\mathbf{x}$ are finite. Suppose there are entries of $\mathbf{x}$ which are $-\infty$. We can renumber the variables to get $\mathbf{x}$ in the form

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ \hline -\infty \\ \vdots \\ -\infty \end{pmatrix} \text{ where } x_1, \ldots, x_k \text{ are finite, and there are } \ell \text{ entries of } -\infty .$$

Then by examining the equation $A\mathbf{x} = \lambda \otimes \mathbf{x}$, we see that *A* must have the form

$$A = \left(\begin{array}{ccc|c} & & & \\ \hline -\infty & \cdots & -\infty & \\ \vdots & \ddots & \vdots & \\ -\infty & \cdots & -\infty & \end{array} \right), \text{ where the block of } -\infty's \text{ is } \ell \times k \text{ and } k + \ell = n .$$

If *A* is irreducible, then we should be able to find a path from *j* to *i*. But if we begin at a vertex with index greater than *k*, there are only arcs to other vertices with index also greater than *k*. Thus, this is a contradiction, so *A* is not irreducible. Therefore all the entries of $\mathbf{x}$ must be finite.

Now we return to the equation $A\mathbf{x} = \lambda \otimes \mathbf{x}$, which we can express as

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \lambda \otimes \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

or equivalently,

$$\begin{cases} \max\left(a_{11} + x_1,\, a_{12} + x_2, \ldots, a_{1n} + x_n\right) = \lambda + x_1 \\ \max\left(a_{21} + x_1,\, a_{22} + x_2, \ldots, a_{2n} + x_n\right) = \lambda + x_2 \\ \quad \vdots \qquad \vdots \qquad \vdots \quad \vdots \quad \vdots \qquad \vdots \\ \max\left(a_{n1} + x_1,\, a_{n2} + x_2, \ldots, a_{nn} + x_n\right) = \lambda + x_n \end{cases} .$$

Consider an arbitrary cycle of length $\ell \le n$ in D_A. Without loss of generality, by renumbering the vertices, let this be the cycle $\sigma : v_1 \to v_2 \to \cdots \to v_k \to v_1$ or $\sigma : a_{12}, a_{23}, \ldots, a_{k1}$. Note that necessarily, since $a_{12}, a_{23}, \ldots, a_{k1}$ are arcs of D_A, then $a_{12}, a_{23}, \ldots, a_{k1} \in \mathbb{R}$. We now consider the set of inequalities which is produced using only this cycle:

$$\begin{cases} a_{12} + x_2 \le \lambda + x_1 \\ a_{23} + x_3 \le \lambda + x_2 \\ \quad \vdots \qquad\quad \vdots \\ a_{k1} + x_1 \le \lambda + x_n \end{cases}$$

If we take the sum of these inequalities, we now have the following result:

$$a_{12} + a_{23} + \cdots + a_{k1} \le \ell \cdot \lambda$$

$$\frac{a_{12} + a_{23} + \cdots + a_{k1}}{\ell} \le \lambda$$

$$M(\sigma) \le \lambda$$

Since we arbitrarily chose a cycle from *A*, we now conclude that $\mu(A) \le \lambda$.

Each row equation for $A\mathbf{x} = \lambda \otimes \mathbf{x}$, $\max\left(a_{i1} + x_1, a_{i2} + x_2, \ldots, a_{in} + x_n\right) = \lambda + x_i$, must achieve equality for some *j*, i.e. $a_{ij} + x_j = \lambda + x_i$. Since *A* is strongly connected, and each vertex must have out-degree at least 1, each vertex must have a cycle. This ensures that we will eventually form a cycle, σ, that is made up of equality conditions, yielding the following set of equations:

$$\begin{cases} a_{ij} + x_j = \lambda + x_i \\ a_{jk} + x_k = \lambda + x_j \\ \qquad \vdots \\ a_{*i} + x_i = \lambda + x_* \end{cases}$$

Taking the sum of these equations, we conclude that

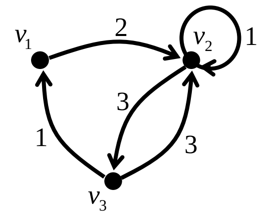
$$\lambda = \frac{a_{ij} + a_{jk} + \cdots + a_{*i}}{\ell_\sigma} = M(\sigma) \le \mu(A).$$

Therefore, $\lambda = \mu(A)$. ■

It will be helpful to consider a numerical example. In example 1.3a, we outline the process for finding the eigenvalue and eigenvector in great detail. In example 1.3b, we give an example to illustrate the case that the digraph is not strongly connected.

Example 1.3a: Finding the Eigenvalue and Eigenvector for an Irreducible Matrix

Let $A = \begin{pmatrix} -\infty & 2 & -\infty \\ -\infty & 1 & 3 \\ 1 & 3 & -\infty \end{pmatrix}$ with D_A :



There are three cycles in A:

$\sigma_1 : (a_{22})$ $\quad M(\sigma_1) = 1/1 = 1$

$\sigma_2 : (a_{23}, a_{32})$ $\quad M(\sigma_2) = (3+3)/2 = 3$

$\sigma_3 : (a_{12}, a_{23}, a_{31})$ $\quad M(\sigma_3) = (2+3+1)/3 = 2$

Thus, $\lambda = \mu(A) = 3$.

To find the eigenvector for $\lambda = 3$, we seek to solve $\begin{pmatrix} -\infty & 2 & -\infty \\ -\infty & 1 & 3 \\ 1 & 3 & -\infty \end{pmatrix} \mathbf{x} = 3 \otimes \mathbf{x}$.

Or equivalently, $\begin{cases} \max(-\infty,\ 2+x_2,\ -\infty) = 3 + x_1 \\ \max(-\infty,\ 1+x_2\ ,\ 3+x_3) = 3 + x_2 \\ \max(1+x_1,\ 3+x_2,\ -\infty) = 3 + x_3 \end{cases}$.

Suppose $x_1 = -\infty$, then it quickly follows that x_2 and x_3 also equal $-\infty$. Since the eigenvector must have at least one entry not equal to $-\infty$, this cannot be. Thus, we let $x_1 = 0$ (recall that eigenvectors are only unique up to max-plus scalar multiples, so we can now let x_1 be any finite value), which results in the following set of equations.

$$\begin{cases} \max(-\infty,\ 2+x_2,\ -\infty) = 3 \\ \max(-\infty,\ 1+x_2\ ,\ 3+x_3) = 3 + x_2 \\ \max(1,\ 3+x_2,\ -\infty) = 3 + x_3 \end{cases}$$

Further, we find that $x_2 = 1$, and reduce the system to two equations:

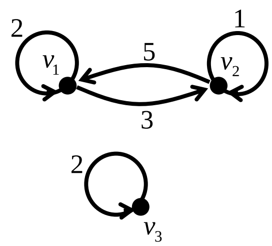
$$\begin{cases} \max(-\infty,\ 2\ ,\ 3+x_3) = 4 \\ \max(1,\ 4,\ -\infty) = 3+x_3 \end{cases}$$

This can be solved for x_3, yielding $x_3 = 1$.

Thus, we have that for $\lambda = 3$, the eigenvector $\mathbf{x}$ can be any $\mathbf{x}$ of the form $\mathbf{x} = c \otimes (0,1,1)^T$, where $c \in \mathbb{R}$. ■

Example 1.3b: Matrix with Multiple Eigenvalues

Let $A = \begin{pmatrix} 2 & 3 & -\infty \\ 5 & 1 & -\infty \\ -\infty & -\infty & 2 \end{pmatrix}$ with D_A :



This matrix has two eigenvalues, $\lambda = 4$ with eigenvector $\mathbf{x}_{\lambda=4} = \begin{pmatrix} 0 & 1 & -\infty \end{pmatrix}^T$ and $\lambda = 2$ with eigenvector $\mathbf{x}_{\lambda=2} = \begin{pmatrix} -\infty & -\infty & 0 \end{pmatrix}^T$. ■

In Example 1.3a, we find only one eigenvector. We can follow up on this by proving that, in fact, in certain instances, this eigenvector is essentially unique.

Theorem 1.3.5: Let *A* be an irreducible $n \times n$ matrix with eigenpair $(\lambda, \mathbf{x})$. If the critical graph of *A* is strongly connected, then the $\mathbf{x}$ is unique (up to max-plus scalar multiples).

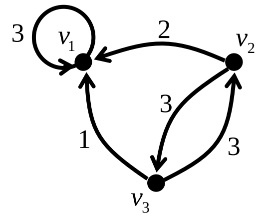
Proof:

Let *A* be an irreducible $n \times n$ matrix such that the critical digraph *D* of *A* is strongly connected. Let $\mathbf{x}$ and $\mathbf{y}$ be eigenvectors of *A* corresponding to $\lambda = \mu(A)$. Since *A* is irreducible, all entries of $\mathbf{x}$ and $\mathbf{y}$ are finite. By using max-plus scaling, we may assume that $x_1 = y_1 = 0$. We show $\mathbf{x} = \mathbf{y}$. Let *j* be any vertex. Since *D* is strongly connected, there exists a path in *D* from i to j: $i = i_1 \to i_2 \to i_3 \to \cdots \to i_k = j$. Since each arc of *D* lies on a cycle of mean $\mu(A)$, $a_{i_\ell, i_{\ell+1}} + x_{i_{\ell+1}} = \lambda + x_{i_\ell}$ and $a_{i_\ell, i_{\ell+1}} + y_{i_{\ell+1}} = \lambda + y_{i_\ell}$. Hence

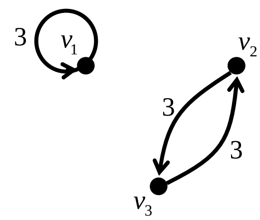
$x_{i_{\ell+1}} - x_{i_\ell} = y_{i_{\ell+1}} - y_{i_\ell}$ (*) for $\ell = 1, \ldots, k-1$. As $x_{i_1} = y_{i_1}$, (*) implies $x_{i_\ell} = y_{i_\ell}$. Therefore $\mathbf{x} = \mathbf{y}$. ■

Example 1.3c: Irreducible Matrix with Multiple Eigenvectors

Let $A = \begin{pmatrix} 3 & 2 & -\infty \\ -\infty & -\infty & 3 \\ 1 & 3 & -\infty \end{pmatrix}$ with D_A :



This matrix has one eigenvalue, $\lambda = 3$. Since the critical graph is not strongly connected (see digraph to right), we expect the possibility of more than one eigenvector. In particular, the eigenvectors for $\lambda = 3$ are $\mathbf{x}_1 = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}^T$ and $\mathbf{x}_2 \in \left\{ (0 \;\; d \;\; d)^T : d \le 1 \right\}$. ■



The most difficult step in calculating the eigenvalue and eigenvectors for an irreducible $n \times n$ matrix, A, is the calculation of $\mu(A)$. Efficient algorithms have been developed for computing $\mu(A)$, including an algorithm with computational complexity $O(n^3)$, which can be found in [6].

Chapter 2: Applications

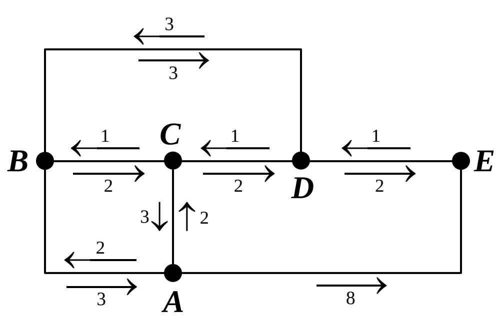
2.1 Shortest Route Problem

Consider a weighted digraph, a system of vertices connected by a collection of (directed) weighted arcs. The arc weight may represent any quantity associated with that arc such as a physical length, time, cost, etc. When examining such a labeling, one goal is to find the shortest, i.e. the most efficient, path from one vertex to another.

This problem can be formulated and solved using the min-plus algebra. Recall that in this algebra, $\oplus$ represents the operation of finding a minimum and ∞ is the additive identity. We can use a matrix, X, to represent this digraph, where each entry x_{ij} represents the smallest arc weight from i to j. If there is more than one arc between i and j, then for the purposes of the shortest route problem, we simply choose the weight of the shortest arc for the entry in the matrix. Note that x_{ij} is not necessarily the same as x_{ji}. If there is no path from i to j, then we assign the value of $x_{ij} = \infty$.

Example: Quickest traffic route
Consider the following map (a digraph) of a road system during rush hour, where the vertices represent road intersections and the weight of each arc actually represents the average time it takes to drive that arc. Notice that most arcs have different times depending on which direction you are driving, the road connecting *B* and *D* is unaffected by the rush hour traffic, and the road from *A* to *E* is one-way. We would like to find the shortest driving time between any two intersections.

D_X :



$$X = \begin{pmatrix} \infty & 2 & 2 & \infty & 8 \\ 3 & \infty & 2 & 3 & \infty \\ 3 & 1 & \infty & 2 & \infty \\ \infty & 3 & 1 & \infty & 2 \\ \infty & \infty & \infty & 1 & \infty \end{pmatrix}$$

The entries of X show the shortest travel times between intersections for one-arc paths. If we examine the i, j entry of the matrix for X^2, it is

$\min\left(x_{i1}+x_{1j}, x_{i2}+x_{2j}, \ldots, x_{in}+x_{nj}\right)$, which gives the shortest two-arc path from i to j. Similarly, X^k holds the shortest driving times for *k*-arc paths between intersections. While searching for the shortest driving route between two intersections, we need only calculate up to X^{n-1} for an $n \times n$ matrix. In this case, we need to find $X^2, X^3,$ and X^4, listed below.

$$X^2 = \begin{pmatrix} 5 & 3 & 4 & 4 & \infty \\ 5 & 3 & 4 & 4 & 5 \\ 4 & 5 & 3 & 4 & 4 \\ 4 & 2 & 5 & 3 & \infty \\ \infty & 4 & 2 & \infty & 3 \end{pmatrix} \quad X^3 = \begin{pmatrix} 6 & 5 & 5 & 6 & 6 \\ 6 & 5 & 5 & 6 & 6 \\ 6 & 4 & 5 & 5 & 6 \\ 5 & 6 & 4 & 5 & 5 \\ 5 & 3 & 6 & 4 & \infty \end{pmatrix} \quad X^4 = \begin{pmatrix} 8 & 6 & 7 & 7 & 8 \\ 8 & 6 & 7 & 7 & 8 \\ 7 & 6 & 6 & 7 & 7 \\ 7 & 5 & 6 & 6 & 7 \\ 6 & 7 & 5 & 6 & 6 \end{pmatrix}$$

Now, to find the shortest driving route from i to j, we need to find the minimum value of the x_{ij} entry in the matrices $X, X^2, X^3,$ and X^4, i.e. we need to find the matrix $X^* = X \,\underline{\oplus}\, X^2 \,\underline{\oplus}\, X^3 \,\underline{\oplus}\, X^4$

$$X^* = \begin{pmatrix} 5 & 3 & 4 & 4 & 6 \\ 5 & 3 & 4 & 4 & 5 \\ 4 & 4 & 3 & 4 & 4 \\ 4 & 2 & 4 & 3 & 5 \\ 5 & 3 & 2 & 4 & 3 \end{pmatrix}$$

Thus, in the matrix X^*, the x_{ij} entry gives the driving time of the shortest path from vertex *i* to vertex *j* [4].

Min-Plus Solution to the Shortest Route Problem:
For a graph with *n* vertices and matrix representation *X* (written in terms of min-plus algebra), let $X^* = X \,\underline{\oplus}\, X^2 \,\underline{\oplus} \ldots \underline{\oplus}\, X^{n-1}$. Then the *ij*-entry of X^* is the weight of the shortest route from vertex i to vertex j.

2.2 The Project Scheduling Problem

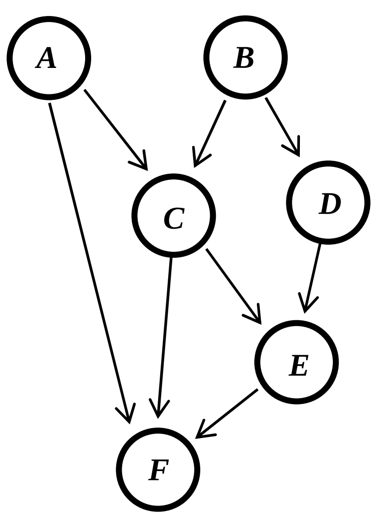
A primary field of importance in Operations Research is that of project scheduling. A complicated project is made up of many tasks that must be accomplished before the project is completed. Some tasks may be carried out simultaneously and others have a precedence order. In project scheduling, we are generally concerned with answering three questions:

1. What is the minimum time in which the project can be completed?
2. Which tasks are the most time-sensitive? If there is a delay in this task, will it cause a delay in the overall production time? If so, then it is considered a bottleneck.
3. Which tasks are the least time-sensitive? If a task is not a bottleneck, then there is some slack time during which delays can occur without disruption to the overall production. For these tasks there will be a critical time, beyond which, delays in the overall production time will occur.

The Project Scheduling Problem has been discussed in other papers [4, 5]; here we present a slightly different formulation of this problem that we think is a little easier to understand.

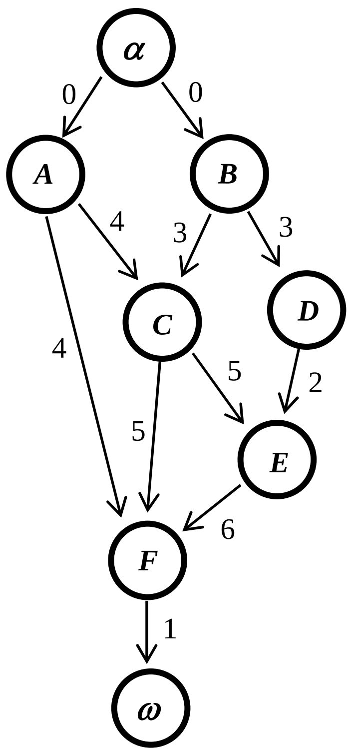
Example: Production Line

Six machines in a production line (*A*, *B*, *C*, *D*, *E*, and *F*) work together to produce a car part. Note that when each machine has gone through its production cycle once, it might produce components that are sent to several other machines. Consider the following task precedence diagram and machine cycle times:



Machine	Time to complete cycle (in minutes)
A	4
B	3
C	5
D	2
E	6
F	1

Every arc in the diagram must be traveled in order for the part to be properly assembled, thus, the minimum time in which the car part can be completed is the time it takes to travel the longest path through the diagram. This should be somewhat reminiscent of the shortest route problem except that we are searching for the maximum path, rather than the minimum path. In order to designate machines with no predecessors and to include the last cycle in the part's completion, it will be helpful to introduce a "Start" and "End" stage to our precedence tree. We denote these the α and ω vertices respectively. If a machine has no predecessors, then the arc weight from α to one of these vertices is 0. If machine X has a cycle completion time t, then all arcs leaving X should have a weight of t. Thus, we can draw a new and more representative precedence diagram that includes all the cycle completion times (see below).



$$X = \begin{array}{c} \alpha \\ A \\ B \\ C \\ D \\ E \\ F \\ \omega \end{array} \left(\begin{array}{cccc|cccc} -\infty & 0 & 0 & -\infty & -\infty & -\infty & -\infty & -\infty \\ & -\infty & -\infty & 4 & -\infty & -\infty & 4 & -\infty \\ & & -\infty & 3 & 3 & -\infty & -\infty & -\infty \\ & & & -\infty & -\infty & 5 & 5 & -\infty \\ \hline & & & & -\infty & 2 & -\infty & -\infty \\ & & & & & -\infty & 6 & -\infty \\ & & & & & & -\infty & 1 \\ & & & & & & & -\infty \end{array}\right)$$

Since the purpose of this problem is to find the maximum path weight through the diagram, we will formulate the matrix representation, X (shown above), in terms of max-plus. Thus, any arc that is not shown on the diagram will be assigned a weight of $-\infty$. Since the digraph has no cycles, the representation matrix can be taken to be upper triangular. We have added partition lines to the matrix X to make it easier to read for calculation purposes.

Just like the solution to the shortest path problem, we need to calculate $X^* = X \overline{\oplus} X^2 \overline{\oplus} \ldots \overline{\oplus} X^{n+1}$ to find the longest path for this problem, where *n* is the number of machines (remember that we added two extra vertices to the diagram). To answer the question of the minimum time to produce the car part, we will really only be concerned with the $x^*_{\alpha\omega}$ entry, which will give us the maximum path weight from α (start) to ω (end). The matrices $X, X^2, X^3, \ldots X^7$ and X^* are given below.

$$X = \begin{array}{c} \alpha \\ A \\ B \\ C \\ D \\ E \\ F \\ \omega \end{array} \left(\begin{array}{cccc|cccc} -\infty & 0 & 0 & -\infty & -\infty & -\infty & -\infty & -\infty \\ & -\infty & -\infty & 4 & -\infty & -\infty & 4 & -\infty \\ & & -\infty & 3 & 3 & -\infty & -\infty & -\infty \\ & & & -\infty & -\infty & 5 & 5 & -\infty \\ \hline & & & & -\infty & 2 & -\infty & -\infty \\ & & & & & -\infty & 6 & -\infty \\ & & & & & & -\infty & 1 \\ & & & & & & & -\infty \end{array}\right)$$

$$X^2 = \begin{array}{c} \alpha \\ A \\ B \\ C \\ D \\ E \\ F \\ \omega \end{array} \left(\begin{array}{cccc|cccc} -\infty & -\infty & -\infty & 4 & 3 & -\infty & 4 & -\infty \\ & -\infty & -\infty & -\infty & -\infty & 9 & 9 & 5 \\ & & -\infty & -\infty & -\infty & 8 & 8 & -\infty \\ & & & -\infty & -\infty & -\infty & 11 & 6 \\ \hline & & & & -\infty & -\infty & 8 & -\infty \\ & & & & & -\infty & -\infty & 7 \\ & & & & & & -\infty & -\infty \\ & & & & & & & -\infty \end{array}\right)$$

$$X^3 = \begin{array}{c} \alpha \\ A \\ B \\ C \\ D \\ E \\ F \\ \omega \end{array} \left(\begin{array}{cccc|cccc} -\infty & -\infty & -\infty & -\infty & -\infty & 9 & 9 & 5 \\ & -\infty & -\infty & -\infty & -\infty & -\infty & 15 & 10 \\ & & -\infty & -\infty & -\infty & -\infty & 14 & 9 \\ & & & -\infty & -\infty & -\infty & -\infty & 12 \\ \hline & & & & -\infty & -\infty & -\infty & 9 \\ & & & & & -\infty & -\infty & -\infty \\ & & & & & & -\infty & -\infty \\ & & & & & & & -\infty \end{array}\right)$$

$$X^4 = \begin{array}{c} \alpha \\ A \\ B \\ C \\ D \\ E \\ F \\ \omega \end{array} \left(\begin{array}{cccc|cccc} -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & 15 & 10 \\ & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & 16 \\ & & -\infty & -\infty & -\infty & -\infty & -\infty & 15 \\ & & & -\infty & -\infty & -\infty & -\infty & -\infty \\ \hline & & & & -\infty & -\infty & -\infty & -\infty \\ & & & & & -\infty & -\infty & -\infty \\ & & & & & & -\infty & -\infty \\ & & & & & & & -\infty \end{array}\right)$$

$$X^5 = \begin{array}{c} \alpha \\ A \\ B \\ C \\ D \\ E \\ F \\ \omega \end{array} \left(\begin{array}{cccc|cccc} -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & 16 \\ & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ & & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ & & & -\infty & -\infty & -\infty & -\infty & -\infty \\ \hline & & & & -\infty & -\infty & -\infty & -\infty \\ & & & & & -\infty & -\infty & -\infty \\ & & & & & & -\infty & -\infty \\ & & & & & & & -\infty \end{array}\right)$$

$$X^6, X^7 = (-\infty)$$

$$X^* = \begin{array}{c} \alpha \\ A \\ B \\ C \\ D \\ E \\ F \\ \omega \end{array} \left(\begin{array}{cccc|cccc} -\infty & 0 & 0 & 4 & 3 & 9 & 15 & \mathbf{16} \\ & -\infty & -\infty & 4 & -\infty & 9 & 15 & 16 \\ & & -\infty & 3 & 3 & 8 & 14 & 15 \\ & & & -\infty & -\infty & 5 & 11 & 12 \\ \hline & & & & -\infty & 2 & 8 & 9 \\ & & & & & -\infty & 6 & 7 \\ & & & & & & -\infty & 1 \\ & & & & & & & -\infty \end{array}\right)$$

The bold entry, $x^*_{\alpha\omega}$, denotes the shortest production time for the part.

From the $x^*_{\alpha\omega}$ entry, we can see that the longest path from start to finish for the production process takes 16 minutes to complete. We cannot produce the part in any smaller time frame since every arc in the graph must be traveled.

There were two other questions concerning time-sensitivity that we wished to answer. To answer these, we need to calculate the longest path through each machine and then use this to calculate the available slack time for each machine. The longest path weight that contains machine i is given by the sum of the longest path from “Start” to i and the longest path from i to “End”, or the expression $x^*_{\alpha i} \otimes x^*_{i\omega}$. Thus we can find a vector of longest paths through each machine, $\mathbf{v}$, where $v_i = x^*_{\alpha i} \otimes x^*_{i\omega}$. We then define a slack vector, $\mathbf{s}$, which is the minimum overall production time less the longest path through each machine, where $s_i = x^*_{\alpha\omega} - v_i$.

$$\mathbf{v} = \begin{pmatrix} 0+16 \\ 0+15 \\ 4+12 \\ 3+9 \\ 9+7 \\ 15+1 \end{pmatrix} = \begin{pmatrix} 16 \\ 15 \\ 16 \\ 12 \\ 16 \\ 16 \end{pmatrix} \qquad \mathbf{s} = \begin{pmatrix} 16-16 \\ 16-15 \\ 16-16 \\ 16-12 \\ 16-16 \\ 16-16 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 4 \\ 0 \\ 0 \end{pmatrix} \qquad \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array}$$

For the purposes of this paper, we will define a **bottleneck** as any machine (or task) for which the slack is zero. Any delay at a machine that is a bottleneck will cause the overall production process to be delayed as well. For this problem, machines *A*, *C*, *E*, and *F* are bottlenecks. The least time-sensitive machines are those with large slack values – in this case, machine *D* has four minutes of slack. However, even those machines with slack time have critical start times. The longest path from machine *D* to completion requires 9 minutes (this is the value of $x^*_{D\omega}$). If the production process starts at time zero, then short delays (less than four minutes) at machine *D* can only occur in the first seven minutes of the production run without causing on overall production delay. We will define the **critical time** to be the time after which any delay in this particular task will delay the overall process. The critical time for each machine, c_i, is simply the overall minimum production time less the longest route from i to "End", or $c_i = x^*_{\alpha\omega} - x^*_{i\omega}$.

$$\mathbf{c} = \begin{pmatrix} 16-16 \\ 16-15 \\ 16-12 \\ 16-9 \\ 16-7 \\ 16-1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 4 \\ 7 \\ 9 \\ 15 \end{pmatrix} \quad \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix}$$

Max-Plus Solution to the Project Scheduling Problem:

1. Formulate a new precedence diagram for the project with new vertices for α and ω. For any task i that has no predecessors, then assign a directional arc from α to i with weight zero. For any task j that has no successors, then assign a directional arc from j to ω with weight equal to the completion time of task j. Then, for any other non-terminating task k, assign all arcs leaving k a weight equal to the completion time of task *k*.
2. Write the Max-plus matrix representation, X, for the graph where the x_{ij} entry is the arc length from task i to task j. If there is no arc connecting i and j, then the x_{ij} entry will be $-\infty$. If the vertices are labeled appropriately, X should be upper triangular with all diagonal entries equal to $-\infty$.
3. Find $X^* = X \overline{\oplus} X^2 \overline{\oplus} \cdots \overline{\oplus} X^{n+1}$, where *n* is the number of vertices before the α and ω vertices were added.
4. The **shortest overall completion time** for the project is $x^*_{\alpha\omega}$.
5. Find the longest route vector ($\mathbf{v}$), the slack vector ($\mathbf{s}$), and the critical time vector ($\mathbf{c}$). These are all $n \times 1$ vectors, they do not include the α or ω vertices.

$$v_i = x^*_{\alpha i} \otimes x^*_{i\omega} \qquad s_i = x^*_{\alpha\omega} - v_i \qquad c_i = x^*_{\alpha\omega} - x^*_{i\omega}$$

6. The **bottlenecks** are the tasks with slack time equal to zero ($s_i = 0$). If the slack time for task i is greater than zero, then there can be delays in this task of time $\le s_i$ that will not effect the overall completion time as long as the delays occur before the critical time, c_i.

2.3 Synchronized Events Problem

The Synchronized Event Problem is similar to the Project Scheduling Problem in that we want to schedule events to meet some deadline. However, the twist is that 1) the events run simultaneously (instead of sequentially) and 2) we want the completion of the longest event to occur exactly at the deadline. This type of situation will often occur with very time-sensitive deadlines. For example, the coordination of system checks for a Space Shuttle Launch, the preparation of a plane for a set takeoff time, or the preparation of an athlete before an Olympic event.

If we are only coordinating the events of a single deadline, then we can find the latest start times by simply taking the difference of the finish time and individual event duration times. For example, when an unloaded plane is brought to its new gate, it will need refueling, maintenance, food service, and luggage service. Suppose these events require times of 20 min, 30 min, 15 min, and 15 min, respectively, and that the plane is supposed to taxi to the runway in 45 minutes. Then taking the difference shows that the latest starting time for each event is as follows: refueling, 25 min; maintenance, 15 min; food service, 30 min; and luggage service, 30 min.

When we need to coordinate similar events for multiple deadlines, then we will only be concerned with timing the maximum event duration with the deadline. For example, consider the case where we now have three planes that arrive at their new gates (*A*, *B*, and *C*) ready for pre-flight preparation. Each plane has different time requirements for refueling and food service (related to the mileage of the next trip), maintenance (depending on whether there were problems on a previous flight or the age of the plane), and luggage service (related to both the mileage of the next trip and the number of passengers on the flight). When on of the pre-flight maintenance teams is sent out, they service all three planes at once (we assume here that there are enough people on each maintenance team to accomplish this). The pre-flight preparation matrix is shown below (event times are in minutes).

$$\begin{array}{cc} \begin{array}{cccc} R & M & F & L \end{array} & \\ \begin{pmatrix} 25 & 10 & 35 & 15 \\ 15 & 45 & 15 & 20 \\ 25 & 15 & 20 & 15 \end{pmatrix} & \begin{array}{l} \text{Gate 1} \\ \text{Gate 2} \\ \text{Gate 3} \end{array} \end{array}$$

Note that in this case, it is in the best interest of the airport to load the food and luggage as late as possible. Latest possible loading will ensure that the food will require the least on-board refrigeration time. Likewise, latest loading of luggage will ensure that the greatest amount of passenger luggage will reach the plane before it leaves the airport.

Example 2.3a: Departure times of $d_1 = 45$, $d_2 = 50$, $d_3 = 55$ minutes

We want to find the latest starting times for the procedures *R*, *M*, *F*, and *L* so that the last procedure is completed at the departure time of the plane. This problem can be formulated as the following max-plus matrix equation, where we want to solve for $\mathbf{t}$:

$$\begin{pmatrix} 25 & 10 & 35 & 15 \\ 15 & 45 & 15 & 20 \\ 25 & 15 & 20 & 15 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} = \begin{pmatrix} 45 \\ 50 \\ 55 \end{pmatrix}$$

We can quickly find the discrepancy matrix, D_a, the reduced discrepancy matrix, R_a, and the candidate solution, $\mathbf{t}^*$ (as discussed in 1.3).

$$D_a = \begin{pmatrix} 20 & 35 & 10 & 30 \\ 35 & 5 & 35 & 30 \\ 30 & 40 & 35 & 40 \end{pmatrix} \qquad R_a = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathbf{t}' = \begin{pmatrix} 20 \\ 5 \\ 10 \\ 30 \end{pmatrix}$$

From R_a, we can tell by the all-zero row that there will be no solution to the problem that has been posed. Indeed we can verify this by trying $\mathbf{t}'$ as a possible solution:

$$\begin{pmatrix} 25 & 10 & 35 & 15 \\ 15 & 45 & 15 & 20 \\ 25 & 15 & 20 & 15 \end{pmatrix} \begin{pmatrix} 20 \\ 5 \\ 10 \\ 30 \end{pmatrix} = \begin{pmatrix} 45 \\ 50 \\ \mathbf{45} \end{pmatrix}$$

The bold entry is the one that causes the solution to fail.

Although $\mathbf{t}'$ does not represent a strict solution to the matrix equation, it does not result in a delay of deadline. That is, the plane at Gate 3 will be ready too soon rather than too late. When the candidate solution is not a strict solution to the matrix equation, but it does not result in a delay of any of the deadline, we will refer to this as a **non-ideal solution**.

Example 2.3b: Departure times of $d_1 = 50$, $d_2 = 55$, $d_3 = 45$ minutes

The control tower decides to reschedule the takeoff times of the three planes due to the extensive maintenance requirements of the plane at the second gate. This results in the following matrix equation, discrepancy matrices, and candidate solution.

$$\begin{pmatrix} 25 & 10 & 35 & 15 \\ 15 & 45 & 15 & 20 \\ 25 & 15 & 20 & 15 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} = \begin{pmatrix} 50 \\ 55 \\ 45 \end{pmatrix}$$

$$D_{\text{b}} = \begin{pmatrix} 25 & 40 & 15 & 35 \\ 40 & 10 & 40 & 35 \\ 20 & 30 & 25 & 30 \end{pmatrix} \qquad R_{\text{a}} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \qquad \mathbf{t}' = \begin{pmatrix} 20 \\ 10 \\ 15 \\ 30 \end{pmatrix}$$

From the reduced discrepancy matrix we can see that the candidate solution, $\mathbf{t}'$, is a solution to the matrix equation. In this example, the start times for maintenance and food service are fixed entries (lone-ones in the second and third column of R_{a}). The start times for refueling and luggage could be earlier without affecting the strict solution. Notice that the presence of so many ones in the third row of R_{a} indicates that for the plane at Gate 3, most of the pre-flight procedures (all but food service) are timed to end simultaneously.

2.4 Airport Problem

In section 1.3, we discussed various results involving the eigenvalue and eigenvector for an irreducible matrix. Before we present an application of the eigenvalue and eigenvector, one might ask what kind of meaning the eigenvalue has in a given problem. Thus, we first need to prove the following theorem.

> **Theorem 2.4.1:** Let $A=[a_{ij}]$ be an irreducible $n\times n$ max-plus matrix. Then $\mu(A)=\min_{\mathbf{x}^*}\max_{i,j}(a_{ij}+x_j-x_i)$, where $\mathbf{x}^*$ is the set of all non-negative $n\times 1$ vectors.

Proof:

Since A is irreducible, A has eigenvalue $\mu(A)$ and an eigenvector $\mathbf{x}$ with finite entries. We may scale $\mathbf{x}$ so that every entry is non-negative. Thus, $\max_j(a_{ij}+x_j)=\mu(A)+x_i$ for all i and $\max_{i,j}(a_{ij}+x_j-x_i)=\mu(A)$. Now let $x\in\mathbf{x}^*$, then certainly $\mu(A)\geq\min_{\mathbf{x}^*}\max_{i,j}(a_{ij}+x_i-x_j)$.

Let $m=\max_{i,j}(a_{ij}+x_j-x_i)$.Then for any i,j, $m\leq(a_{ij}+x_j-x_i)$. Now consider a cycle $\sigma: i_1\to i_2\to\ldots\to i_\ell\to i_1$ in D_A with cycle length ℓ. Using this cycle, we have the following system of inequalities:

$$\begin{aligned}
a_{i_1i_2}+x_{i_2}-x_{i_1}&\leq m\\
a_{i_2i_3}+x_{i_3}-x_{i_2}&\leq m\\
\vdots\qquad\qquad&\ \ \vdots\\
a_{i_\ell i_1}+x_{i_1}-x_{i_\ell}&\leq m
\end{aligned}$$

Taking the sum of these inequalities gives

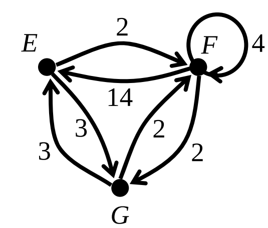
$$\begin{aligned}
&a_{i_1i_2}+a_{i_2i_3}+\cdots+a_{i_\ell i_1}\leq\ell\cdot m\\
&\frac{a_{i_1i_2}+a_{i_2i_3}+\cdots+a_{i_\ell i_1}}{\ell}\leq m\\
&M(\sigma)\leq m
\end{aligned}$$

Since $M(\sigma)\leq m$ for an arbitrary cycle σ, then $\mu(A)\leq m$. Thus $\mu(A)\leq\min_{\mathbf{x}^*}\max_{i,j}(a_{ij}+x_i-x_j)$. ■

Example: Airport Problem

Consider an airline company that manages three rural airports, E, F, and G. Each airport can send flights to or receive flights from the other two airports. An airport may also send and receive the same flight. We let A be the 3×3 matrix where the a_{ij} entry represents the flight time for the flight from j to i (note that this is the transpose of how we would normally set up the matrix representation). If there is no flight from j to i, then $a_{ij} = -\infty$. We define the eigenvector $\mathbf{x}$ as the vector where x_i is the time when airport i will open.

$$A = \begin{pmatrix} -\infty & 14 & 3 \\ 2 & 4 & 2 \\ 3 & 2 & -\infty \end{pmatrix} \qquad D_A:$$



In terms of this problem, we summarize the meaning of individual terms:

$\mathbf{x}$	The vector that contains the airport opening schedule.
$a_{ij} + x_j - x_i$	The time after i opens that the plane from j arrives at i.
$\max\limits_{i,j}\left(a_{ij} + x_j - x_i\right)$	The maximum time an airport needs to be open given the schedule $\mathbf{x}$.
$\min\limits_{\mathbf{x}^*}\max\limits_{i,j}\left(a_{ij} + x_j - x_i\right)$	The shortest **equal** time period that all the airports would need to be open given any possible non-negative schedule $\mathbf{x}$. $(=\lambda)$

Thus, for this problem, the eigenvalue corresponds to the minimum equal time period that all the airports must remain open to ensure that all the planes take off and land appropriately.

Let's find the eigenvalue and an eigenvector for the matrix A. There are six cycles in D_A.

$\sigma_1 : F \to F$ $\qquad M(\sigma_1) = 4/1 = 4$

$\sigma_2 : E \to F \to E$ $\qquad M(\sigma_2) = (14+2)/2 = 8$

$\sigma_3 : F \to G \to F$ $\qquad M(\sigma_3) = (2+2)/2 = 2$

$\sigma_4 : G \to E \to G$ $\qquad M(\sigma_4) = (3+3)/2 = 3$

$\sigma_5 : E \to F \to G \to E$ $\qquad M(\sigma_5) = (3+2+2)/3 = 7/3$

$\sigma_6 : E \to G \to F \to E$ $\qquad M(\sigma_6) = (14+2+3)/3 = 19/3$

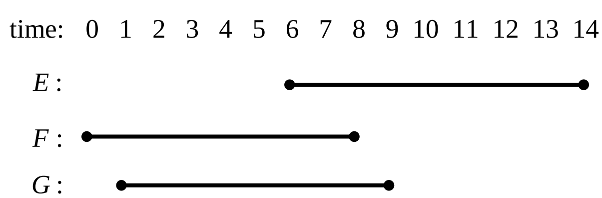
We seek an eigenpair $(8, \mathbf{x})$ such that $A\mathbf{x} = 8 \otimes \mathbf{x}$.

$$\begin{pmatrix} -\infty & 14 & 3 \\ 2 & 4 & 2 \\ 3 & 2 & -\infty \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 8 \otimes \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \qquad \Leftrightarrow \qquad \begin{cases} \max(-\infty,\ 14 + x_2,\ 3 + x_3) = 8 + x_1 \\ \max(2 + x_1,\ 4 + x_2,\ 2 + x_3) = 8 + x_2 \\ \max(3 + x_1,\ 2 + x_2,\ -\infty) = 8 + x_3 \end{cases}$$

Let $x_1 = 0$. Then the system becomes $\begin{cases} \max(-\infty,\ 14 + x_2,\ 3 + x_3) = 8 \\ \max(2,\ 4 + x_2,\ 2 + x_3) = 8 + x_2 \\ \max(3,\ 2 + x_2,\ -\infty) = 8 + x_3 \end{cases}$.

Solving for the remaining components, we have $x_2 = -6$ or $x_3 = -5$.

This gives the eigenvector $\mathbf{x} = (0\ \ -6\ \ -5)^T$. We scale the eigenvector so that the smallest entry is zero to keep from having to use negative times. We can use the max-plus scalar multiple, $\mathbf{x} \otimes 6$, to give us an equivalent eigenvector $\mathbf{x} = (6\ \ 0\ \ 1)^T$. The diagram below shows the time period in which each airport will be open, where each airport will be open for 8 hours.



Note that this schedule does not take into account the possible closings and openings of airports within the 8-hour stretch. It only gives a schedule for when the earliest opening and the latest closing occur. Also, we assume that the airport schedules are dependent on flight scheduling and not vice versa.

Conclusion

In this paper, we have seen that many characteristics of the max-plus algebraic structure are similar to those in more familiar mathematical structures. We can use matrix operations, solve systems of max-plus equations, and have existence and meaning for eigenvalues and eigenvectors.

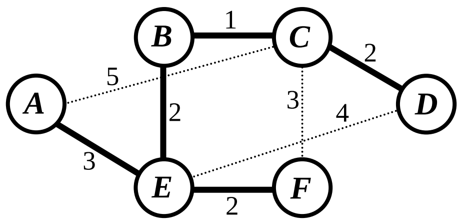
Through applications, we have seen that max-plus and min-plus provide interesting tools that can be used to formulate and solve many problems of optimization. There are numerous applications of max-plus, and they are certainly not limited to those presented in this paper. We have tried to present the topic in an understandable and reader-friendly way and encourage the reader to seek out new applications of the max-plus algebra.

References

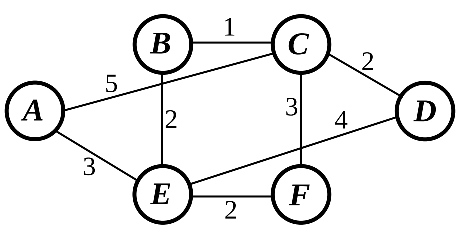
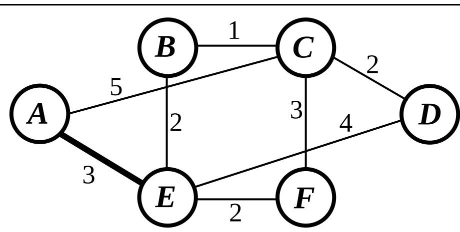
1. R.B. Bapat, D. Stanford, and P. van den Driessche. Pattern properties and spectral inequalities in max algebra. *SIAM Journal of Matrix Analysis and Applications*, 16(3): 964-976, 1995.
2. Norman Biggs. *Algebraic Graph Theory*, 2nd ed. Cambridge University Press, 1993.
3. Peter Butkovic. Strong regularity of matrices – a survey of results. *Discrete Applied Mathematics*, 48:45-68, 1994.
4. R.A. Cuninghame-Green. *Minimax Algebra*. Lecture notes in Economics and Mathematical Systems, 166. Springer 1979.
5. Stephane Gaubert. Methods and Applications of (max,+) Linear Algebra. *INRIA Rocquencourt*, 1997.
6. R.M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Math* 23:309-311 (1978).
7. Don Phillips, A. Ravindran, James J. Solberg. *Operations Research, Principles and Practice*. John Wiley & Sons, 1976.
8. Peter Tannenbaum and Robert Arnold. *Excursions in Modern Mathematics, 3rd ed.* Prentice Hall, 1992, p.254.

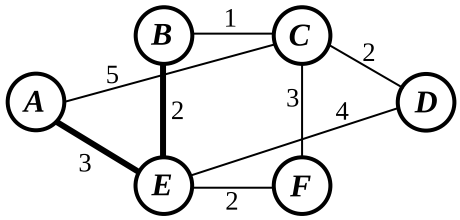
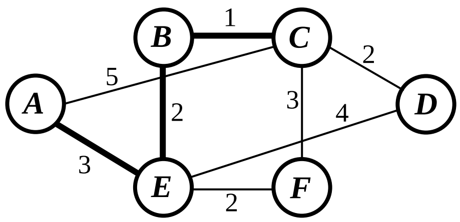
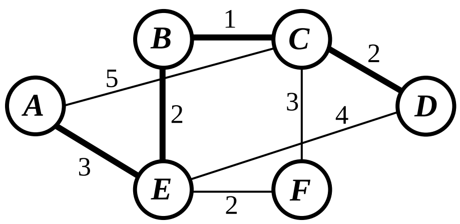
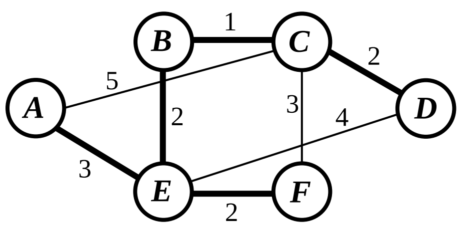
Appendix: The Minimum Spanning Tree Problem

Given an undirected graph with weighted edges, a **spanning tree** is a set of edges that include every vertex on the graph. A **minimum spanning tree** (MST) is a spanning tree with the smallest total edge length. For example, in a communications network, we often want to find the most efficient way to connect several locations. For illustration, a minimum spanning tree for the graph below has been marked with solid lines.



There are several algorithmic procedures that may be used to find the minimum spanning tree for a given problem. Here we present Prim's algorithm [8], which always results in a minimum spanning tree. Begin at any vertex of the graph and find the shortest edge out of that vertex (this is the first vertex in the MST). Then find the shortest edge out of either of the vertices in the MST that goes to an unincluded vertex. Add this edge and vertex to the tree. Again, find the shortest edge out of any of the included vertices in the MST that goes to an unincluded vertex. Add this edge and vertex to the tree. Repeat this process until all the vertices have been included in the minimum spanning tree. It is important to understand the algorithmic approach on a graph before we present the min-plus procedure, so we have outlined the use of Prim's algorithm for finding the MST of the above graph in the following table.

<table>
<tr><td>
</td><td>The original graph. Choose a vertex to begin with, in this example, we begin with vertex A.</td></tr>
<tr><td>
</td><td>The shortest edge out of vertex A is path AE with length 3. So we choose this edge to begin the MST.</td></tr>
</table>

	Now we look for the shortest edge out of either *A* or *E* that will connect to *B*, *C*, *D*, or *F*. There are two candidates, EB or EF. Either will work; we include EB.
	Now we look for the shortest edge out of either *A*, *E*, or *B* that will connect to *C*, *D*, or *F*. We choose edge BC, which has a length of 1.
	Now we look for the shortest edge out of either *A*, *E*, *B*, or *C* that will connect to *D* or *F*. There are two candidate edges of length 2, CD and EF. We include CD.
	Finally, we have only one more vertex to include, vertex *F*. The shortest edge to *F* from the MST-in-progress is EF. This completes the minimum spanning tree.

The procedure is not difficult, but it can be quite difficult to ensure that you have considered all the possible edges to non-included vertices as the number of vertices on the graph increases. For this reason, we present a methodical and quick approach to finding the minimum spanning tree using the min-plus algebra. Please note that in problems where we seek to find a minimum spanning tree for a graph, it is necessary that edges include both directions equally, that is $x_{ij} = x_{ji}$ for all i, j. In addition, when we write the matrix representation for the graph, *X*, we do not include arcs that are loops since these would never be considered in the MST. Thus, the x_{ii} entries will always be ∞ entries. Likewise, if i and j are not connected by an edge, then $x_{ij} = x_{ji} = \infty$.

$$X = \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{pmatrix} \infty & \infty & 5 & \infty & 3 & \infty \\ \infty & \infty & 1 & \infty & 2 & \infty \\ 5 & 1 & \infty & 2 & \infty & 3 \\ \infty & \infty & 2 & \infty & 4 & \infty \\ 3 & 2 & \infty & 4 & \infty & 2 \\ \infty & \infty & 3 & \infty & 2 & \infty \end{pmatrix}$$

We now outline a min-plus algorithm, similar to Prim's algorithm, for finding the minimum spanning tree given the min-plus MST matrix representation of the graph.

Min-Plus Solution to find the Minimum Spanning Tree:

Choose a row, i, to begin the process.

Mark the ith row and ith column as included (using 1's).

Repeat the following three steps until all the rows and columns are marked as included.

1. Choose (circle) an entry, x_{jk}, that remains in one of the included rows, which is equal to the minimum value of all the entries in the included rows. Also circle the x_{kj} entry.
2. The jth row and column will already be marked as included. Now mark the kth row and column as included (using 1's).
3. Cross out (using **X**) any entry that is the intersection entry of an included row and column.

The circled entries of the matrix X give the arcs that are included in the minimum spanning tree. Note that a given graph may have more than one minimum spanning tree.

In the min-plus algorithm, we mark rows and columns as "included" so that we keep a record of which vertices have already been included in the minimum spanning tree. Once a new vertex has been included, we no longer need to consider any edges that connect the new vertex to any of the included vertices. Thus, the included row / column intersections are removed (crossed out) at the end of each repetition. Although all the work to find the MST using the min-plus algorithm can be done on a single copy of the matrix X, we will show the procedure step by step through to completion on separate matrices.

Example: Finding the Minimum Spanning Tree using the Min-plus Solution

We will find the MST for $X = \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array}\begin{pmatrix} \infty & \infty & 5 & \infty & 3 & \infty \\ \infty & \infty & 1 & \infty & 2 & \infty \\ 5 & 1 & \infty & 2 & \infty & 3 \\ \infty & \infty & 2 & \infty & 4 & \infty \\ 3 & 2 & \infty & 4 & \infty & 2 \\ \infty & \infty & 3 & \infty & 2 & \infty \end{pmatrix}$. The procedure follows.

We begin with the arbitrary choice of row 1 and mark the 1st row and 1st column as included.	$X = \begin{array}{c\|cccccc} & 1 & & & & & \\ \hline 1 & \infty & \infty & 5 & \infty & 3 & \infty \\ & \infty & \infty & 1 & \infty & 2 & \infty \\ & 5 & 1 & \infty & 2 & \infty & 3 \\ & \infty & \infty & 2 & \infty & 4 & \infty \\ & 3 & 2 & \infty & 4 & \infty & 2 \\ & \infty & \infty & 3 & \infty & 2 & \infty \end{array}$
Now we circle a minimum entry in the included rows (in this step, the only included row is the first one). The minimum entry of row 1 is x_{15}, so we circle this entry and the x_{51} entry and mark the 5th row and column as included.	$X = \begin{array}{c\|cccccc} & 1 & & & & 1 & \\ \hline 1 & \infty & \infty & 5 & \infty & 3 & \infty \\ & \infty & \infty & 1 & \infty & 2 & \infty \\ & 5 & 1 & \infty & 2 & \infty & 3 \\ & \infty & \infty & 2 & \infty & 4 & \infty \\ 1 & 3 & 2 & \infty & 4 & \infty & 2 \\ & \infty & \infty & 3 & \infty & 2 & \infty \end{array}$
Before we repeat the process, we need to cross out any new intersection entries, in this case, they are x_{11} and x_{55}.	$X = \begin{array}{c\|cccccc} & 1 & & & & 1 & \\ \hline 1 & \cancel{\infty} & \infty & 5 & \infty & 3 & \infty \\ & \infty & \infty & 1 & \infty & 2 & \infty \\ & 5 & 1 & \infty & 2 & \infty & 3 \\ & \infty & \infty & 2 & \infty & 4 & \infty \\ 1 & 3 & 2 & \infty & 4 & \cancel{\infty} & 2 \\ & \infty & \infty & 3 & \infty & 2 & \infty \end{array}$
Now we find a new minimum entry in the included rows (row 1 and row 5). Row 5 contains two minimum entries of 2. Arbitrarily, we choose one of these, x_{52}. Thus we circle x_{52} and x_{25} and mark the 2nd row and column as included.	$X = \begin{array}{c\|cccccc} & 1 & 1 & & & 1 & \\ \hline 1 & \cancel{\infty} & \infty & 5 & \infty & 3 & \infty \\ 1 & \infty & \infty & 1 & \infty & 2 & \infty \\ & 5 & 1 & \infty & 2 & \infty & 3 \\ & \infty & \infty & 2 & \infty & 4 & \infty \\ 1 & 3 & 2 & \infty & 4 & \cancel{\infty} & 2 \\ & \infty & \infty & 3 & \infty & 2 & \infty \end{array}$

Now we cross out the new intersection entries, x_{12}, x_{21}, and x_{22}.	$X = \begin{array}{ccccccc} & 1 & 1 & & & 1 & \\ 1 & \cancel{\infty} & \cancel{\infty} & 5 & \infty & 3 & \infty \\ 1 & \cancel{\infty} & \cancel{\infty} & 1 & \infty & 2 & \infty \\ & 5 & 1 & \infty & 2 & \infty & 3 \\ & \infty & \infty & 2 & \infty & 4 & \infty \\ 1 & 3 & 2 & \infty & 4 & \cancel{\infty} & 2 \\ & \infty & \infty & 3 & \infty & 2 & \infty \end{array}$
We find a new minimum entry in the included rows (rows 1, 2, and 5), circling x_{23} and x_{32}. We mark the 3rd row and column as included and then cross out the new intersection entries, x_{31}, x_{13}, x_{33}, x_{35}, and x_{53}.	$X = \begin{array}{ccccccc} & 1 & 1 & 1 & & 1 & \\ 1 & \cancel{\infty} & \cancel{\infty} & \cancel{5} & \infty & 3 & \infty \\ 1 & \cancel{\infty} & \cancel{\infty} & 1 & \infty & 2 & \infty \\ 1 & \cancel{5} & 1 & \cancel{\infty} & 2 & \cancel{\infty} & 3 \\ & \infty & \infty & 2 & \infty & 4 & \infty \\ 1 & 3 & 2 & \cancel{\infty} & 4 & \cancel{\infty} & 2 \\ & \infty & \infty & 3 & \infty & 2 & \infty \end{array}$
We find the new minimum entry in the included rows (rows 1, 2, 3, and 5), arbitrarily choosing x_{34} (x_{56} would have also been an acceptable choice). Entries x_{34} and x_{43} are circled and the 4th row and column are marked as included. The new intersection entries are crossed out.	$X = \begin{array}{ccccccc} & 1 & 1 & 1 & 1 & 1 & \\ 1 & \cancel{\infty} & \cancel{\infty} & \cancel{5} & \cancel{\infty} & 3 & \infty \\ 1 & \cancel{\infty} & \cancel{\infty} & 1 & \cancel{\infty} & 2 & \infty \\ 1 & \cancel{5} & 1 & \cancel{\infty} & 2 & \cancel{\infty} & 3 \\ 1 & \cancel{\infty} & \cancel{\infty} & 2 & \cancel{\infty} & \cancel{4} & \infty \\ 1 & 3 & 2 & \cancel{\infty} & \cancel{4} & \cancel{\infty} & 2 \\ & \infty & \infty & 3 & \infty & 2 & \infty \end{array}$
This must be the last step since there was only one unincluded column/row at the end of the previous step. The last choice for a minimum entry in an included row is x_{56}. Thus x_{56} and x_{65} are circled and the remaining uncircled entries are crossed out.	$X = \begin{array}{ccccccc} & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \cancel{\infty} & \cancel{\infty} & \cancel{5} & \cancel{\infty} & 3 & \cancel{\infty} \\ 1 & \cancel{\infty} & \cancel{\infty} & 1 & \cancel{\infty} & 2 & \cancel{\infty} \\ 1 & \cancel{5} & 1 & \cancel{\infty} & 2 & \cancel{\infty} & \cancel{3} \\ 1 & \cancel{\infty} & \cancel{\infty} & 2 & \cancel{\infty} & \cancel{4} & \cancel{\infty} \\ 1 & 3 & 2 & \cancel{\infty} & \cancel{4} & \cancel{\infty} & 2 \\ 1 & \cancel{\infty} & \cancel{\infty} & \cancel{3} & \cancel{\infty} & 2 & \cancel{\infty} \end{array}$
This gives us the minimum spanning tree for the graph with matrix X.	$MST_X = \begin{pmatrix} \infty & \infty & \infty & \infty & 3 & \infty \\ \infty & \infty & 1 & \infty & 2 & \infty \\ \infty & 1 & \infty & 2 & \infty & \infty \\ \infty & \infty & 2 & \infty & \infty & \infty \\ 3 & 2 & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & 2 & \infty \end{pmatrix}$