OPTIMIZING AND APPROXIMATING EIGENVECTORS IN MAX-ALGEBRA

By

KIN PO TAM

A thesis submitted to
The University of Birmingham
for the Degree of
DOCTOR OF PHILOSOPHY (PHD)

School of Mathematics
The University of Birmingham
March, 2010

Abstract

This thesis is a reflection of my research in max-algebra. The idea of max-algebra is replacing the conventional pairs of operations $(+, \times)$ by $(\max, +)$.

It has been known for some time that max-algebraic linear systems and eigenvalue-eigenvector problem can be used to describe industrial processes in which a number of processors work interactively and possibly in stages. Solutions to such max-algebraic linear system typically correspond to start time vectors which guarantee that the processes meet given deadlines or will work in a steady regime.

The aim of this thesis is to study such problems subjected to additional requirements or constraints. These include minimization and maximization of the time span of completion times or starting times. We will also consider the case of minimization and maximization of the time span when some completion times or starting times are prescribed.

The problem of integrality is also studied in this thesis. This is finding completion times or starting times which consists of integer values only. Finally we consider max-algebraic permuted linear systems where we permute a given vector and decide if the permuted vector is a satisfactory completion time or starting time.

For some of these problems, we developed exact and efficient methods. Some of them turn out to be hard. For these we have proposed and tested a number of heuristics.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Aims and Scopes of the Thesis

In this thesis we will introduce the concepts of max-plus algebra and the results of my research in this topic will be presented.

The idea of max-algebra is replacing the conventional pairs of operations $(+, \times)$ by $(\max, +)$. It has been known for some time that max-algebraic linear systems and eigenvalue-eigenvector problem can be used to describe industrial processes in which a number of processors work interactively and possibly in stages. Solutions to such max-algebraic linear system typically correspond to start time vectors which guarantee that the processes meet given deadlines or will work in a steady regime.

The aim of this thesis is to study such problems subjected to additional requirements or constraints. We will start by introducing the basic notation and results regarding max-plus algebra, max-algebraic linear system and eigenvalue-eigenvector problem. Using these results, we can study the problem of minimization and maximization of the time span of completion times or starting times. We will also study the case of minimization and maximization of the time span when some completion times or starting times are prescribed. We will show that the above problems can be solvable by exact and efficient method.

Next we will present results on integer max-algebraic linear system where we investigate if integer completion times or starting times exists. We will show that for some special cases, this problem can be solved efficiently by checking necessary and sufficient conditions. An algorithm is also developed which provides a benchmark for solving the general case.

Finally we will study the max-algebraic permuted linear system problem. This is to find if there exists a permutation on a given completion times or starting times vector such that it is a solution to the max-algebraic linear systems or eigenvalue-eigenvector problem. It turns out this problem is NP-complete but we will develop efficient algorithms for solving the case when the problem is small. We will also propose and test a number of heuristics for solving this problem.

1.2 Literature Review

In this section we will discuss the historical background of max-plus algebra and some of the works that were done on this topic.

The idea of max-plus algebra was first seen in the 1950s or even at an earlier period. But this topic was not given too much attention at the time and the theories started to develop by the 1960s. Works were developed by the Operations Research community including Cuninghame-Green [31], Romanovski [56] and Vorobyov [63]. One of the first detailed publications on max-plus algebra was the 'Minimax algebra' by Cuninghame-Green [29]. An updated version of this book was later published in 1995 [30].

In the first publications, max-algebraic linear systems were investigated [29], [31], [63]. These include the systems of the forms $A \otimes x = b$, $A \otimes x = x \oplus b$ and $A \otimes x = x$ where $\oplus = \max$ and $\otimes = +$. Cuninghame-Green first published a column maxima method which solves the problem $A \otimes x = b$ for a given A and b in his 1960 paper [31]. This result has also been found independently by Zimmermann [66]. Cuninghame-Green [29] later published another method for solving this problem by using residuation [11]. This method required to first consider the inequality $A \otimes x \leq b$ and obtain the maximal solution from this inequality. It was proved that this maximal solution can be obtained by finding

$$\bar{x}_j = \min_i (-a_{ij} + b_i)$$

and the system $A \otimes x = b$ has a solution if and only if $A \otimes \bar{x} = b$. Akian, Gaubert and Kolokoltsov [2] have extended the set covering theorem to infinite dimension. Also Gaubert [37] proposed a method for solving the one-sided system $x = A \otimes x \oplus b$ by using rational calculus.

During the period of the 1970s and 1980s a lot of new technologies were developed especially in manufacturing. With more complex systems being built, the synchronization of discrete event (dynamic) systems (DES or DEDS) which include machine scheduling, queueing and network process etc, became more important. Publications including [29] and [67] give a detail interpretation on this field. 'Synchronization and Linearity' by Baccelli, Cohen, Olsder, Quadrat [6] provide a detail account on deterministic system theory and stochastic DES. It was shown that the synchronization problems can be formulated as the max-algebraic eigenvalue-eigenvector problem [6], [29].

The eigenvalue-eigenvector problem is to find an eigenvalue λ and an eigenvector x such that $A \otimes x = \lambda \otimes x$ for a given matrix A. It was shown that in machine scheduling, if we use an eigenvector as the starting times for the machines then the system will reach what we call a steady state where each machine will have the same cycle duration λ. Cuninghame-Green [29] shows that the max-algebraic eigenvalue-eigenvector problem is related to the longest distances problem by converting A into a directed graph D_A. He shows that for any matrix

A, its maximum cycle mean of the directed graph D_A, namely $\lambda(A)$ where

$$\lambda(A) = \max_{\sigma} \frac{w(\sigma)}{l(\sigma)} \tag{1.1}$$

plays an important role in solving the max-algebraic eigenvalue-eigenvector problem. In (1.1), σ represents any cyle in A, $w(\sigma)$ is the weight of the cycle and $l(\sigma)$ is the length of the cycle. He proved that the greatest eigenvalue called the principal eigenvalue is equal to the maximum cycle mean. It was also shown in [29] that if a matrix A is irreducible then $\lambda(A)$ is the unique eigenvalue and A has only finite eigenvectors.

Various algorithms were developed for finding the eigenvalue of an irreducible matrix. Karp's algorithm [44] has remained one of the fastest and most commonly used algorithms for finding the maximum cycle mean with computational complexity $O(mn)$ where m is the number of edges in the associated directed graph. Dasdan and Gupta provide a fast algorithm for finding maximum and minimum cycle mean but no computational complexity was given [26]. Elsner and van den Driessche [34] developed a power algorithm of computational complexity $O(n^4)$ and this was later modified based on Karp's algorithm in [35]. Cuninghame-Green developed an algorithm by conversion to linear programming for irreducible matrices in [29].

General max-algebraic two-sided linear systems have been investigated extensively, results can be found in [22], [32], [33] and [64]. A general solution method was presented by Walkup and Borriello [64]. This method uses for its basic solution component, the max-plus closure operation and solves a series of subsystems with decreasing maximum solution.

An elimination method for solving $A \otimes x = B \otimes x$ was presented by Butkovič and Hegedus [22]. It was also shown that the solution set is generated by a finite number of vectors.

Cuninghame-Green and Zimmermann [33] developed a general iterative approach which assumes that finite upper and lower bounds for all variables be given. The iterative method makes it possible to find an approximation of the maximum solution to the given system, which satisfies the given lower and upper bounds or to find out that no such solution exists.

A pseudo-polynomial algorithm called Alternating method for solving $A \otimes x = B \otimes y$ was presented by Cuninghame-Green and Butkovič in [32]. The algorithm converges to a finite solution from any finite starting point whenever a finite solution exists or finds out that no solution exists. Sergeev [58] extended the Alternating method to the generalized systems

$$A^1 \otimes x^1 = ... = A^k \otimes x^k.$$

It was also shown that if all the input entries are real, the Alternating method finds a finite solution in a finite number of steps or decides that no solution exists for the given problem.

Butkovič [14] has shown that there is strong relation between max-algebra and combinatorics. Comprehensive results regarding strongly regularity and the relation between the image set, simple image set and the eigenspace can also be found in [14].

Probabilistic max-plus algebra which is motivated by dynamic programming and large deviations have been developed by Akian, Quadrat and Viot in [4], by Del Moral and Salut [51], [52].

Butkovič [16] has investigated the permuted max-algebraic eigenvector problem. This is to find out if it is possible to permute the components of a given vector x so that it becomes an eigenvector of $A \in \overline{\mathbb{R}}^{n \times n}$. It was shown that this problem is NP-complete by using a polynomial transformation from BANDWIDTH in [36]. In [16] it was shown that for any given matrix $A \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^m$ the permuted linear system problem is also NP-complete. That is to find if there exists a permutation π such that after applying the permutation to b then $b(\pi)$ is an image of A, i.e. if $A \otimes x = b(\pi)$ has a solution.

The theories of max-plus algebra have also been applied to many different areas, this includes combinatorial optimization [14], functional Galois connection [2], agriculture [5], biological sequences [25] and a railway system [43].

1.3 Motivation of the Problem

From the previous section we can see that max-plus algebra has been applied in many different areas. One of the problems attracting interests has been the machine scheduling problem. We will illustrate this problem by considering the following example.

Example 1.3.1. [29] Consider a manufacturer that has n machines, $M_1, ..., M_n$, working interactively and in stages. A machine cannot start a new stage before it receives the components (parts) produced by other machines at the previous stage.

Suppose that M_j will require a_{ij} units of time to produce component P_i which is necessary for machine M_i in the next stage for $i, j = 1, ..., n$ and that each machine can produce all components simultaneously. The entry $a_{ij} = -\infty$ if M_i does not need a component from M_j. We will assume that the manufacturer would want each machine to start a new stage again as soon as it finishes its process so that they can produce as many components as possible. The system of machines working interactively in this way is called a *multi-machine interactive production system.*

The manufacturer may want to find the time they should start/switch on each machine over a period of time, i.e the starting time of each machine at the first stage. Therefore the manufacturers may wish to model this situation mathematically so that the starting times of each machine can be decided according to their preferences.

We will denote by $x_j(1)$ the starting time of machine M_j for $j = 1, ..., n$ when it is started for the first time. Similarly we will denote $x_j(k)$ to be the starting time of machine M_j when it is started k^{th} time. We will also denote

$$x(k) = (x_1(k), x_2(k), ..., x_n(k))$$

to be the vector of starting times of the individual machines at the k^{th} stage. Therefore the 2^{nd} starting time for machine M_i will be

$$x_i(2) = \max\{a_{i1} + x_1(1), ..., a_{in} + x_n(1)\}.$$

Similarly the $k+1^{th}$ starting time for machine M_i will be

$$x_i(k+1) = \max\{a_{i1} + x_1(k), ..., a_{in} + x_n(k)\}. \tag{1.2}$$

In order to decide the starting times for each machine, the manufacturers may wish to formulate this problem in what we call a max-plus algebra linear system. Namely, if we denote $\oplus = \max$, $\otimes = +$ then (1.2) becomes

$$x(k+1) = A \otimes x(k). \tag{1.3}$$

Since every machine is likely to perform the same task repeatedly over a period of time, at each stage a machine will start and finish the process and wait until the next starting time. One of the criteria the manufacturers may want to meet when choosing the starting times is that the starting times between two consecutive stages differ by the same constant for every machine. This can be modelled as a max-plus algebra eigenvalue-eigenvector problem. That is $x(k+1) = \lambda \otimes x(k)$ or by (1.3) equivalently

$$A \otimes x(k) = \lambda \otimes x(k).$$

Hence if we choose a max-algebraic eigenvector as the vector of starting times for their machines then the system will immediately reach a *steady state* which means that all of their machines will have the same cycle duration. However this requirement usually does not determine the starting times vector uniquely because the eigenspace may have many independent eigenvectors. In real-life systems there is a choice of eigenvectors and it may be desirable to set up some other criteria on deciding which one of those eigenvectors is more suitable for the manufacturers to use as the vector of starting times for their machines.

1.4 Overview of Chapters

The main aim of this research is to consider some of the criteria which may be set by the manufacturer. Then we will develop methods for finding eigenvectors which satisfied these criteria. We will divide our results into different chapters and a brief overview of each chapter are as follows:

Chapter 2: **Introducing Max-Plus Algebra System**

In this chapter we will provide the terminology, notations and basic definitions of max-algebra. We will also present some of the theories on linear systems and using them to define the notions of the image set and the simple image set which will play a significant role in this thesis.

Chapter 3: **Max-algebraic Eigenvalues and Eigenvectors**

In this chapter, we will present definitions and some of the well known results on the max-algebraic eigenvalue-eigenvector problem. We will first discuss the concept of a steady state and how it is related to the max-algebraic eigenproblem. Then we will show that graph theory and the max-algebraic eigenproblem are very much related. Using this relation, we will present a solution method for finding all eigenvalues and eigenvectors for any square matrices A. We will also show that for each eigenvalue, we can obtain the set of eigenvectors by considering the image set of a matrix generated from A. Therefore we will transform the problem of optimizing eigenvectors into optimizing the image set of a matrix.

Chapter 4: **Optimizing Range Norm of the Image Set**

In this chapter, we will consider the problem of minimizing and maximizing range norm of an image set of a matrix. We will first consider the minimization problem and we will investigate the case when the image set is finite and the case when the image set may not be finite. Then we will move on to the maximization problem and obtain a solution method for this case.

Chapter 5: **Optimizing Range Norm of the Image Set With Prescribed Components**

This chapter will be an extension of Chapter 4. We will investigate a similar problem as in Chapter 4 but in this case, we include an additional constraint to the problem. This additional constraint will be part of the image vector is prescribed and therefore fixed. We will develop a solution method for finding the non-prescribed components such that the resulting image vector have its range norm minimized or maximized.

Chapter 6: **Integer Linear System**

In this chapter, we will investigate the case of integer linear system problem. We will first consider the case when the matrix only consists of one and two columns and find the conditions for an integer image set not to be non-empty. We will then move on to the case of strongly regular matrices and we will do the same as for the case of one and two columns matrices. Finally we will investigate the general case.

Chapter 7: **On Permuted Linear Systems**

In this chapter, we will investigate the permuted linear system problem which is NP-complete. We will develop algorithms to decide if a permuted vector is in the image of A for the case of $n = 2$, $n = 3$ and $n > 3$. We will also for the case $n = 1$ and $n = 2$, develop algorithms to find out if we can find a permuted vector such that the distance from this vector to the image set is minimized where this distance is measured by using the Chebyshev norm.

Chapter 8: **Heuristics for the Permuted Linear Systems Problem**

Finally in this chapter, we will develop heuristic methods to obtain an approximation of the solution to the permuted linear systems problem. We will compare the running time and results by testing all the heuristics methods we develop in this chapter.

Note that the results in Chapters 2 and 3 have been intensively studied before. These results can be found in [2], [14], [29], [30], [31], [63] and [66]. The results in Chapter 4, 5, 6, 7 and 8 are all original. Also note that the results in Chapter 4 and 7 have been published and they can be found in [23]. Finally results in Chapter 5 and 6 are new and have been obtained after the submission of MPhil(Qual) [62].

Chapter 2

Introducing Max-Plus Algebra System

2.1 Introduction

Let us start by introducing the concept of max-plus algebra. In this chapter we will define the basic concepts and definitions of max-plus algebra for scalars, then we will show how they can be extended to matrices and vectors.

In max-plus algebra we use the operation '$\max$' which stands for maximum and the operation '$+$' which stands for addition, to replace the binary operations of addition and multiplication in conventional linear algebra, respectively.

We will use the operators $\oplus$ and $\otimes$ (circle over the conventional $+$ and $\times$) to represent that we are considering the max-plus algebraic system rather than the conventional linear algebraic system.

2.2 Basic Concepts and Definitions

Definition 2.2.1. *Max-plus algebra* is the linear algebra over the semiring $\overline{\mathbb{R}}$ (or $\mathbb{R}_{\max}$) where $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty\}$, equipped with the operations of addition '$\oplus = \max$' and multiplication '$\otimes = +$'. The identity element for addition (*zero*) is $-\infty$ and the identity element for multiplication (*unit*) is 0.

Definition 2.2.2. *Min-plus algebra* (or *tropical algebra*) is linear algebra over the semiring $\mathbb{R}_{\min}$ where $\mathbb{R}_{\min} = \mathbb{R} \cup \{+\infty\}$, equipped with the operations of addition '$\oplus' = \min$' and multiplication '$\otimes' = +$'. The identity element for addition (*zero*) is $+\infty$ and the identity element for multiplication (*unit*) is 0.

The completed max-plus semiring $\overline{\overline{\mathbb{R}}}_{max}$ is the set $\overline{\overline{\mathbb{R}}} = \mathbb{R} \cup \{\pm\infty\}$, equipped with the operations of addition '$\oplus = \max$' and multiplication '$\otimes = +$'. We have $-\infty$ is the zero element in max-plus algebra, therefore we will have that

$$+\infty \otimes' -\infty = -\infty = -\infty \otimes' +\infty.$$

The completed min-plus semiring $\overline{\overline{\mathbb{R}}}_{\min}$ is defined in the dual way. Note that this means it is the set $\overline{\overline{\mathbb{R}}} = \mathbb{R} \cup \{\pm\infty\}$, equipped with the operations of addition '$\oplus' = \min$' and multiplication '$\otimes' = +$'. We have $+\infty$ is the zero element in min-plus algebra in which we will have that

$$+\infty \otimes -\infty = +\infty = -\infty \otimes +\infty.$$

Formally, let $a, b \in \mathbb{R}$ then

$$\begin{aligned} a \oplus b &= \max(a, b), \\ a \otimes b &= a + b, \\ a \oplus' b &= \min(a, b), \\ a \otimes' b &= a + b. \end{aligned}$$

Note that the element $+\infty$ only appears when using certain techniques, i.e. dual operations and conjugation. In this thesis we will only develop theory of max-plus algebra over $\overline{\mathbb{R}}$ and we do not attempt to develop a concise max-plus algebraic theory over $\overline{\overline{\mathbb{R}}}$.

For simplicity we will use the word *max-algebra* to represent max-plus algebra for the rest of this thesis. For convenience we will define the symbol ϵ to stand for $-\infty$ and we will also denote by the same symbol any vector or matrix whose every component is $-\infty$.

2.2.1 Algebraic Properties of Max-Algebra

Using the definition of the operators $\oplus$ and $\otimes$, $\forall a, b, c \in \overline{\mathbb{R}}$ the following basic algebraic properties of max-algebra can be easily deduced.

$a \oplus b = \max(a, b)$	$a \otimes b = a + b$
$a \oplus b = b \oplus a$	$a \otimes b = b \otimes a$
$a \oplus (b \oplus c) = (a \oplus b) \oplus c$	$a \otimes (b \otimes c) = (a \otimes b) \otimes c$
$a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$	
$a \oplus \epsilon = a = \epsilon \oplus a$	$a \otimes 0 = a = 0 \otimes a$
	$a \otimes \epsilon = \epsilon = \epsilon \otimes a$
	$(\forall a \in \mathbb{R})\ (\exists a^{-1}) a \otimes a^{-1} = 0$

Table 2.1: Basic Algebraic Properties

Using the definition of $\oplus$ and $\otimes$ again, $\forall a, b, c \in \overline{\mathbb{R}}$ the following properties for inequalities in max-algebra can also be easily deduced.

$$a \leq b \Longrightarrow a \oplus c \leq b \oplus c,$$

$$a \leq b \Longleftrightarrow a \otimes c \leq b \otimes c, \quad c \in \mathbb{R},$$

$$a \leq b \Longleftrightarrow a \oplus b = b.$$

For addition and multiplication of matrices and vectors in max-algebra, the operators $\oplus$ and $\otimes$ can be applied similarly as in the conventional linear algebra. They are used as follows:

For real matrices $A = (a_{ij}),\ B = (b_{ij})$ of compatible sizes, $\alpha \in \mathbb{R}$:

$$A \oplus B = (a_{ij} \oplus b_{ij}),$$
$$A \otimes B = (\sum_{k}^{\oplus} a_{ik} \otimes b_{kj}),$$
$$\alpha \otimes A = (\alpha \otimes a_{ij}).$$

Example 2.2.1.

$$\begin{pmatrix} 2 & 4 \\ -1 & 5 \end{pmatrix} \oplus \begin{pmatrix} -1 & 7 \\ 0 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 7 \\ 0 & 5 \end{pmatrix}.$$

Example 2.2.2.

$$\begin{pmatrix} 2 & 3 \\ -1 & 4 \end{pmatrix} \otimes \begin{pmatrix} 5 \\ 2 \end{pmatrix} = \begin{pmatrix} 7 \\ 6 \end{pmatrix}.$$

Example 2.2.3.

$$3 \otimes \begin{pmatrix} 7 & -3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 10 & 0 \\ 5 & 4 \end{pmatrix}.$$

Using the definition of the operators $\oplus$ and $\otimes$ for matrices and vectors, then $\forall A, B, C \in \overline{\mathbb{R}}^{m \times n}$ of compatible sizes we can obtain the following properties for operations over matrices and vectors in max-algebra.

$A \oplus B = B \oplus A$	
$A \oplus (B \oplus C) = (A \oplus B) \oplus C$	$A \otimes (B \otimes C) = (A \otimes B) \otimes C$
$A \otimes (B \oplus C) = A \otimes B \oplus A \otimes C$	$(A \oplus B) \otimes C = A \otimes C \oplus B \otimes C$

Table 2.2: Properties for operations over matrices and vectors

It immediately follows that for all matrices A, B, C, vectors x, y of compatible sizes and $\alpha, \beta \in \overline{\mathbb{R}}$, then

$$\begin{aligned}
A \otimes (\alpha \otimes B) &= \alpha \otimes (A \otimes B), \\
\alpha \otimes (A \oplus B) &= \alpha \otimes A \oplus \alpha \otimes B, \\
x^T \otimes \alpha \otimes y &= \alpha \otimes (x^T \otimes y), \\
(\alpha \oplus \beta) \otimes A &= \alpha \otimes A \oplus \beta \otimes A, \\
A \leq B &\Longrightarrow A \oplus C \leq B \oplus C, \\
A \leq B &\Longrightarrow A \otimes C \leq B \otimes C, \\
x \leq y &\Longrightarrow A \otimes x \leq A \otimes y, \\
A \leq B &\Longleftrightarrow A \oplus B = B.
\end{aligned} \tag{2.1}$$

Also if A is a square matrix, then the iterated products

$$A \otimes A \otimes ... \otimes A$$

in which the matrix A is multiplied by k times will be denoted by A^k.

Definition 2.2.3. Let $a, b, c, ...$ be real numbers then the matrix:

$$\text{diag}(a, b, c, \dots) = \begin{pmatrix} a & & & & \\ & b & & \epsilon & \\ & & c & & \\ & \epsilon & & \ddots & \\ & & & & \ddots \end{pmatrix}$$

is called a *diagonal* matrix.

For any given vector $d = (d_1, d_2, ..., d_n)$, diag$(d_1, d_2, ..., d_n)$ will be denoted by diag(d).

Max-algebra identity matrix is a diagonal matrix with all diagonal entries equal to zero. Therefore using the definition of the diagonal matrices, we can obtain the following definition.

Definition 2.2.4. Let $I = \text{diag}(0, ..., 0)$ then in max-algebra the matrix I is called the *identity matrix*.

By the definition of the identity matrix, it immediately implies that $I \otimes A = A = A \otimes I$ for any matrices A and I of compatible sizes. By definition $A^0 = I$ for any square matrix.

Definition 2.2.5. Let $a, b \in \mathbb{R}$, then b is called *inverse* of a if

$$a \otimes b = 0 = b \otimes a$$

and we will denote $b = a^{-1}$.

Similarly the concept of an inverse can be applied into matrices.

Definition 2.2.6. Let $A \in \overline{\mathbb{R}}^{n \times n}$, then A is called *invertible* if there exists $B \in \overline{\mathbb{R}}^{n \times n}$ s.t.

$$A \otimes B = I = B \otimes A.$$

It has been proved in [29] that if the matrix B exists, it is unique and we will denote $B = A^{-1}$.

Any matrix which can be obtained from the identity (diagonal) matrix by permuting the rows and/or columns is called a *permutation matrix* (*generalized permutation matrix*) [14]. The position of generalized permutation matrices is slightly more special in max-algebra than in conventional linear algebra as they are the only matrices having an inverse:

Theorem 2.2.1. [29] Let $A \in \overline{\mathbb{R}}^{n \times n}$, then a matrix $B \in \overline{\mathbb{R}}^{n \times n}$ such that

$$A \otimes B = I = B \otimes A$$

exists if and only if A is a generalized permutation matrix.

Example 2.2.4. Let

$$A = \begin{pmatrix} \epsilon & 2 & \epsilon \\ \epsilon & \epsilon & -4 \\ 1 & \epsilon & \epsilon \end{pmatrix} \text{ and } B = \begin{pmatrix} \epsilon & \epsilon & -1 \\ -2 & \epsilon & \epsilon \\ \epsilon & 4 & \epsilon \end{pmatrix}.$$

Then

$$A \otimes B = \begin{pmatrix} \epsilon & 2 & \epsilon \\ \epsilon & \epsilon & -4 \\ 1 & \epsilon & \epsilon \end{pmatrix} \otimes \begin{pmatrix} \epsilon & \epsilon & -1 \\ -2 & \epsilon & \epsilon \\ \epsilon & 4 & \epsilon \end{pmatrix} = \begin{pmatrix} 0 & \epsilon & \epsilon \\ \epsilon & 0 & \epsilon \\ \epsilon & \epsilon & 0 \end{pmatrix} = I$$

and

$$B \otimes A = \begin{pmatrix} \epsilon & \epsilon & -1 \\ -2 & \epsilon & \epsilon \\ \epsilon & 4 & \epsilon \end{pmatrix} \otimes \begin{pmatrix} \epsilon & 2 & \epsilon \\ \epsilon & \epsilon & -4 \\ 1 & \epsilon & \epsilon \end{pmatrix} = \begin{pmatrix} 0 & \epsilon & \epsilon \\ \epsilon & 0 & \epsilon \\ \epsilon & \epsilon & 0 \end{pmatrix} = I.$$

Therefore B is an inverse of A.

Given $\alpha \in \mathbb{R}$, then the *conjugate* of α is $\alpha^* = \alpha^{-1}$. This is equivalent to $-\alpha$ in conventional notation. Similarly we have $(-\infty)^* = +\infty$ and $(+\infty)^* = -\infty$. Given a matrix $A = (a_{ij}) \in \overline{\overline{\mathbb{R}}}^{m \times n}$, the *transpose* of A is denoted by $A^T = (a_{ji})$. Using this, we can obtain the following definition for the conjugate of a matrix.

Definition 2.2.7. Let $A = (a_{ij}) \in \overline{\overline{\mathbb{R}}}^{m \times n}$, then the *conjugate of the matrix* A is $A^* = (a^*_{ji})$. This is obtained by the negation and transposition of the matrix A, i.e. $A^* = -A^T$ in conventional notation.

For simplicity, we will denote the columns (rows) of $A = (a_{ij}) \in \overline{\overline{\mathbb{R}}}^{m \times n}$ by $A_1, ..., A_n$ $(a_1, ..., a_m)$. Also, we will call a matrix or a vector *finite* if none of its entries is $-\infty$ or $+\infty$.

Furthermore, we will mostly work with the matrices which have at least one finite entry on each row or/and column. We will define these matrices by the following definition.

Definition 2.2.8. [29] Let $A \in \overline{\overline{\mathbb{R}}}^{m\times n}$ be a matrix which has at least one finite entry on each row (column) then A is called *row* $\mathbb{R}$*-astic* (*column* $\mathbb{R}$*-astic*). A is called *doubly* $\mathbb{R}$*-astic* if it is both row and column $\mathbb{R}$-astic.

2.3 Max-Algebraic Linear System

In this section we will consider the max-algebraic linear systems of equations and inequalities, namely the system $A \otimes x = b$ and $A \otimes x \leq b$. For simplicity we will use the following notation

$$M = \{1, ..., m\}, N = \{1, ..., n\},$$

where m and n are given positive integers.

2.3.1 System of Linear Equations

Let us first consider the system of linear equations in max-algebra. Given $A = (a_{ij}) \in \overline{\mathbb{R}}^{m\times n}$ and $b = (b_1, ..., b_m) \in \overline{\mathbb{R}}^{m}$, then the system

$$A \otimes x = b \tag{2.2}$$

is called the *one sided max-algebraic linear system* or *max-linear system.* Using conventional notation, the above system can be written as follows:

$$\max_{j=1,...,n} (a_{ij} + x_j) = b_i, \quad i \in M.$$

Now if we subtract the value of b_i for all i from both sides of the equation then we will have

$$\max_{j=1,\dots,n}(a_{ij} - b_i + x_j) = 0, \quad i \in M.$$

If we let the matrix $\bar{A} = (\bar{a}_{ij}) = (a_{ij} - b_i)$, then we will obtain a new system where the right-hand side of the system is equal to 0, i.e.

$$\bar{A} \otimes x = 0.$$

Now we will say that the system is *normalized* and this process is called *normalization*. Note that if we let

$$B = \mathrm{diag}(-b_1, -b_2, \dots, -b_m) = \begin{pmatrix} -b_1 & & & & \\ & -b_2 & & \epsilon & \\ & & & & \\ & \epsilon & & \ddots & \\ & & & & -b_m \end{pmatrix},$$

then

$$B \otimes A \otimes x = \bar{A} \otimes x = 0.$$

Therefore we can see that normalization is equivalent to the multiplication of the system by the matrix B from the left and we can obtain the matrix $\bar{A} = B \otimes A$.

Example 2.3.1. Suppose we have the following system

$$\begin{pmatrix} 7 & -3 & 2 \\ 2 & 1 & 4 \\ 6 & 3 & 9 \\ 4 & 2 & 8 \end{pmatrix} \otimes \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 7 \\ 9 \end{pmatrix}.$$

Then

$$B = \begin{pmatrix} 1 & \epsilon & \epsilon & \epsilon \\ \epsilon & -3 & \epsilon & \epsilon \\ \epsilon & \epsilon & -7 & \epsilon \\ \epsilon & \epsilon & \epsilon & -9 \end{pmatrix}$$

and

$$\bar{A} = \begin{pmatrix} 1 & \epsilon & \epsilon & \epsilon \\ \epsilon & -3 & \epsilon & \epsilon \\ \epsilon & \epsilon & -7 & \epsilon \\ \epsilon & \epsilon & \epsilon & -9 \end{pmatrix} \otimes \begin{pmatrix} 7 & -3 & 2 \\ 2 & 1 & 4 \\ 6 & 3 & 9 \\ 4 & 2 & 8 \end{pmatrix} = \begin{pmatrix} 8 & -2 & 3 \\ -1 & -2 & 1 \\ -1 & -4 & 2 \\ -5 & -7 & -1 \end{pmatrix}.$$

Therefore after normalization we will have

$$\begin{pmatrix} 8 & -2 & 3 \\ -1 & -2 & 1 \\ -1 & -4 & 2 \\ -5 & -7 & -1 \end{pmatrix} \otimes \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Now if we consider the above example we can see that after normalization we will have the following four equations:

$$\begin{aligned}
\max(\quad 8 + x_1, \quad -2 + x_2, \quad 3 + x_3) \quad &= 0, \\
\max(\quad -1 + x_1, \quad -2 + x_2, \quad 1 + x_3) \quad &= 0, \\
\max(\quad -1 + x_1, \quad -4 + x_2, \quad 2 + x_3) \quad &= 0, \\
\max(\quad -5 + x_1, \quad -7 + x_2, \quad -1 + x_3) \quad &= 0.
\end{aligned}$$

Now let us consider the first equation. If (x_1, x_2, x_3) is a solution to the problem, then we know that

$$8 + x_1 \leq 0, -2 + x_2 \leq 0, 3 + x_3 \leq 0$$

or

$$x_1 \leq -8, x_2 \leq 2, x_3 \leq -3.$$

We also know that at least one of these inequalities must be satisfy will equality. Now if we only consider x_1 then from the four equations, we will have obtained that

$$x_1 \leq -8, x_1 \leq 1, x_1 \leq 1, x_1 \leq 5.$$

Therefore $x_1 \leq \min(-8, 1, 1, 5) = -\max(8, -1, -1, -5) = \bar{x}_1$ where $-\bar{x}_1$ is the column maximum of the first column.

Similarly for x_2 and x_3 we can obtain $\bar{x}_2$ and $\bar{x}_3$ which are the column maxima for the second and third column respectively. From above we know that at least one of the inequalities must be satisfied with equality for each equation. It implies that if (x_1, x_2, x_3) is a solution to the problem, then there exists at least one column maximum in each row of the normalized matrix. This has been observed in [2], [14], [31], [63], [66].

Suppose we have the system $A \otimes x = b$ where $A = (a_{ij}) \in \overline{\mathbb{R}}^{m \times n}$ and $b = (b_1, ..., b_m) \in$

$\overline{\mathbb{R}}^m$ are given, then we denote

$$S(A,b) = \{x \in \overline{\mathbb{R}}^n \mid A \otimes x = b\},$$
$$M_j(A,b) = \{k \in M \mid (a_{kj} - b_k) = \max_{i=1,\ldots,m}(a_{ij} - b_i)\}, \ \ \forall j \in N,$$
$$\bar{x}_j = -\max_{i=1,\ldots,m}(a_{ij} - b_i), \ \ \forall j \in N.$$

Let us first consider the case when $b = \epsilon$. This immediately implies that

$$S(A,b) = \{x \in \overline{\mathbb{R}}^n \mid x_j = \epsilon \text{ if } A_j \neq \epsilon, j \in N\}.$$

Therefore if we have $A = \epsilon$, then $S(A,b) = \overline{\mathbb{R}}^n$. Now if we consider the case when $A = \epsilon$ and $b \neq \epsilon$, then we can deduce that $S(A,b) = \emptyset$. Therefore we will assume that $A \neq \epsilon$ and $b \neq \epsilon$.

Now let us suppose that $b_k = \epsilon$ for some $k \in M$, then for any $x \in S(A,b)$ we have $x_j = \epsilon$ if $a_{kj} \neq \epsilon$, $j \in N$. This means that the k^{th} equation of the system can be removed. As a result we can set $x_j = \epsilon$ for every column A_j where $a_{kj} \neq \epsilon$ (if any) and these columns can be removed from the system. Therefore we can assume without loss of generality, that b is finite.

Furthermore, if b is finite and A contains an ϵ row, then it immediately follows that $S(A,b) = \emptyset$. Similarly, if A contains an ϵ column, i.e. $A_j = \epsilon$ for some $j \in N$, then we can set x_j to be any value in a solution x. Therefore without loss of generality, we can suppose that A is doubly $\mathbb{R}$-astic.

Theorem 2.3.1. [66] Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and $b \in \mathbb{R}^m$. Then $x \in S(A,b)$ if and only if

i) $x \leq \bar{x}$ and

ii) $\bigcup_{j \in N_x} M_j(A,b) = M$ where $N_x = \{j \in N \mid x_j = \bar{x}_j\}$.

It follows immediately that the system (2.2) has a solution if and only if $\bar{x}$ is a solution to the system and $\bar{x}$ is called the *principal solution* of the system [29]. More precisely the following two corollaries are observed in [14]:

Corollary 2.3.1. [14] Let $A \in \overline{\mathbb{R}}^{m \times n}$ be doubly $\mathbb{R}$-astic and $b \in \mathbb{R}^m$. Then the following three statements are equivalent:

i) $S(A,b) \neq \emptyset$,

ii) $\bar{x} \in S(A,b)$,

iii) $\bigcup_{j \in N} M_j(A,b) = M$.

Corollary 2.3.2. [14] Let $A \in \overline{\mathbb{R}}^{m \times n}$ be doubly $\mathbb{R}$-astic and $b \in \mathbb{R}^m$. Then $S(A,b) = \{\bar{x}\}$ if and only if

i) $\bigcup_{j \in N} M_j(A,b) = M$ and

ii) $\bigcup_{j \in N'} M_j(A,b) \neq M$ for any $N' \subseteq N, N' \neq N$.

Note that Corollary 2.3.1 provides the conditions for the solvability of the system $A \otimes x = b$ and Corollary 2.3.2 provides the conditions for the existence of a unique solution in the system $A \otimes x = b$.

From the above two corollaries, we can see that solvability and unique solvability of the system $A \otimes x = b$ are equivalent to the set covering and minimal set covering respectively

[14].

2.3.2 System of Linear Inequalities

Now we will consider the system of inequalities. Given $A = (a_{ij}) \in \overline{\mathbb{R}}^{m\times n}$ and $b = (b_1, ..., b_m) \in \overline{\mathbb{R}}^m$, then the system

$$A \otimes x \leq b \tag{2.3}$$

is called *one-sided max-linear system of inequalities* or just *max-linear system of inequalities*. The system of linear inequalities (2.3) have been investigated in the past [29], [66]. It turns out that the system will always have a solution and it can be easily solved. A solution set can be found by considering the following theorem.

Theorem 2.3.2. [28][66] Suppose $A \in \overline{\overline{\mathbb{R}}}^{m\times n}$, $b \in \overline{\overline{\mathbb{R}}}^m$ and $x \in \overline{\overline{\mathbb{R}}}^n$, then

$$A \otimes x \leq b \text{ if and only if } x \leq A^* \otimes' b.$$

It follows from the definition of the principal solution that $\bar{x} = A^* \otimes' b$ if A is doubly $\mathbb{R}$-astic and b is finite. Therefore we will extend this definition and call $A^* \otimes' b$ to be the *principal solution* to the system (2.2) and (2.3) for any $A \in \overline{\mathbb{R}}^{m\times n}$ and $b \in \overline{\mathbb{R}}^m$.

From the above theorem, it immediately follows that the principal solution is the greatest solution to the system $A \otimes x \leq b$.

2.3.3 Image Set

Now let us continue on looking at the max-linear system of equations, namely (2.2). Given $A \in \overline{\mathbb{R}}^{m\times n}$, then $b \in \overline{\mathbb{R}}^m$ is called an *image* of A if there exists $x \in \overline{\mathbb{R}}^n$ such that

$$A \otimes x = b.$$

We will call the set of all images of A to be the *image set* of A and it is defined by the following definition.

Definition 2.3.1. Let $A \in \overline{\mathbb{R}}^{m\times n}$ then $Im(A) = \{A \otimes x \mid x \in \overline{\mathbb{R}}^{n}\}$ is the image set of A.

Using the definitions and results we have discussed in the previous sections, we obtain the following trivial proposition.

Proposition 2.3.1. Given $A \in \overline{\mathbb{R}}^{m\times n}$ and $b \in \overline{\mathbb{R}}^{m}$, then

$$b \in Im(A) \text{ if and only if } S(A, b) \neq \emptyset.$$

Using the above statement, we can find out if a vector is an image of any matrix. From Corollary 2.3.1, we have obtained a combinatorial method to solve the system of linear equations and hence checking if a vector is an image of a matrix. By using Theorem 2.3.2, we can produce an algebraic method for forming this check .

Corollary 2.3.3. Let $A \in \overline{\mathbb{R}}^{m\times n}$ and $b \in \overline{\mathbb{R}}^{m}$ then

$$b \in Im(A) \text{ if and only if } A \otimes (A^{*} \otimes' b) = b.$$

Proof. " $\Rightarrow$ " Let $b \in Im(A)$, then $\exists x$ s.t. $A \otimes x = b$. By Theorem 2.2.1, we have $x \leq A^{*} \otimes' b$. After we have multiply (from the left) both sides of the inequalities by A, we will get

$$b = A \otimes x \leq A \otimes (A^{*} \otimes' b) \leq b$$

and hence $A \otimes (A^{*} \otimes' b) = b$. " $\Leftarrow$ " Let $x = (A^{*} \otimes' b)$, then by the definition of the image set, $b \in Im(A)$. $\square$

Definition 2.3.2. Let $S \subseteq \overline{\mathbb{R}}^{n}$, if $\forall x, y \in S$ and $\forall \alpha \in \overline{\mathbb{R}}$,

$$\alpha \otimes x \in S \text{ and } x \oplus y \in S,$$

then S is called a *max-algebraic subspace* or briefly, a subspace of $\overline{\mathbb{R}}^n$.

Note that from (2.1), it follows that for any $x, y \in \overline{\mathbb{R}}^n$, $\alpha, \beta \in \mathbb{R}$,

$$A \otimes (\alpha \otimes x \oplus \beta \otimes y) = \alpha \otimes A \otimes x \oplus \beta \otimes A \otimes y$$

and hence we can obtain the following proposition:

Proposition 2.3.2. Let $A \in \overline{\mathbb{R}}^{m \times n}$, $\alpha, \beta \in \mathbb{R}$ and $u, v \in Im(A)$, then

$$\alpha \otimes u \oplus \beta \otimes v \in Im(A).$$

By the above proposition, we can deduce that $Im(A)$ is a subspace.

2.3.4 Strongly Regular Matrices and Simple Image Set

Now we shall look at a special kind of matrix in max-algebra. Let us first consider the following vectors $A_1, A_2, ..., A_n \in \overline{\mathbb{R}}^m$, then the vectors are said to be *linearly dependent* [15] if one of these vectors can be expressed as a linear combination of the others, i.e. $\exists k \in N$, $x_1, ..., x_{k-1}, x_{k+1}, ..., x_n \in \mathbb{R}$ such that

$$A_k = \sum_{i \in N - \{k\}}^{\oplus} x_i \otimes A_i.$$

The vectors are said to be *linearly independent* [15] if they are not linearly dependent. Furthermore, we say that these vectors are *strongly linearly independent* if there exists some $b \in \mathbb{R}^m$ s.t. it can be uniquely expressed as a linear combination of $A_1, A_2, ..., A_n$, i.e. the system

$$\sum_{j \in N}^{\oplus} A_j \otimes x_j = b$$

has a unique solution. Also, if $m = n$, then the matrix $A = (A_1, A_2, ..., A_n)$ is called *strongly regular* [15]. It was proved in [29] that strongly linearly independent vectors are also linearly independent.

Definition 2.3.3. [14] Let $A \in \overline{\mathbb{R}}^{m\times n}$ then

$$S_A = \{b \in \mathbb{R}^m \mid A \otimes x = b \text{ has a unique solution}\}$$

is called the *simple image set* of A.

In other words, the simple image set of A is the set of vectors b for which the system (2.2) has a unique solution.

By the definition of the simple image set, it is immediate that the simple image set is a subset of the image set, i.e. $S_A \subseteq Im(A)$. Also by the definition of strongly regular matrix and the simple image set, we see that $S_A \neq \emptyset$ if and only if A has strongly linear independent columns.

Due to the fact that regularity and linearly independence are closely related to the number of solutions of the linear systems (2.2), we will now look at the following results.

Theorem 2.3.3. [15] Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and $b \in \mathbb{R}^m$, then $|S(A,b)| \in \{0, 1, \infty\}$.

From the above theorem, we know that the number of solutions to the linear system can only be 0, 1 and ∞ and this is the same as the conventional case. Now let us suppose that $A \in \overline{\mathbb{R}}^{m\times n}$, then we will denote

$$T(A) = \{|S(A,b)| \mid b \in \mathbb{R}^m\}.$$

Theorem 2.3.4. [15] Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic, then $T(A)$ is either $\{0, \infty\}$ or $\{0, 1, \infty\}$.

We can see that A has strongly linearly independent columns if and only if $1 \in T(A)$.

Next we will show a method on how to check the strongly regularity of a matrix. Suppose that $A \in \overline{\mathbb{R}}^{n\times n}$ is a strongly regular matrix, then by Corollary 2.3.2 there exists $b \in \mathbb{R}^n$ such that the union of the sets

$$M_1(A,b), M_2(A,b), ..., M_n(A,b), \tag{2.4}$$

form a minimal set covering of N. It is not difficult to prove that this can only happen if and only if the sets (2.4) are one-element and pairwise disjoint. That is, for some permutation π of the set N, we have

$$M_{\pi(j)}(A,b) = \{j\}, \ \ \forall j \in N,$$

i.e.

$$a_{j,\pi(j)} \otimes b_j^{-1} > a_{i,\pi(j)} \otimes b_i^{-1}, \ \ \forall i,j \in N \text{ and } i \neq j. \tag{2.5}$$

This is the same as finding if we can multiply constants (or add in the conventional case) to every row of A in a way that there is only one column maximum in every row. Now if we multiply (2.5) over all $j \in N$, we then have for every $\sigma \in P_n - \{\pi\}$

$$\prod_{j\in N}^{\otimes} a_{j,\pi(j)} > \prod_{j\in N}^{\otimes} a_{j,\sigma(j)}. \tag{2.6}$$

We will denote

$$\text{maper}(A) = \sum_{\sigma\in P_n}^{\oplus} \prod_{j\in N}^{\otimes} a_{j,\sigma(j)}$$

or in conventional notational

$$\text{maper}(A) = \max_{\sigma\in P_n} \sum_{j\in N} a_{j,\sigma(j)}.$$

It is called the *max-algebraic permanent* of A and we can see from (2.6) that for a strongly regular matrix, the max-algebraic permanent is uniquely determined by a permutation from P_n.

Looking back at the definition of the max-algebraic permanent in conventional notational, we can see that finding maper(A) is the same as solving the linear assignment problem. There are a number of efficient solution methods for finding an optimal solution, one of the best known is the Hungarian method of computational complexity $O(n^3)$. We will denote the set of all optimal permutations by $ap(A)$, that is

$$ap(A) = \{\pi \in P_n \mid \text{maper}(A) = \sum_{j \in N} a_{j,\pi(j)}\}.$$

If $|ap(A)| = 1$, then we will say A has *strong permanent*; it immediately follows that if A is strongly regular, then A has a strong permanent. The converse is also true for real matrices and hence the following result.

Theorem 2.3.5. [15] Let $A \in \overline{\mathbb{R}}^{m \times n}$ be doubly $\mathbb{R}$-astic, then A is strongly regular if and only A has strong permanent.

Note that we can check if A has strong permanent in $O(n^3)$ time (see [15]).

We have now introduced the basic definitions and concepts of a simple image set and strongly regular matrices. It turns out that the simple image sets are closely related to the max-algebraic eigenvalue-eigenvector problem and this can be seen in [15].

2.4 Summary

In this chapter we have introduced the basic concepts of max-algebra and we have defined its basic properties over matrices and vectors. We have presented some well known results regarding the max-linear system of equations and system of inequalities; we have shown how solution of these systems can be found. We have also introduced the concepts of image set and simple image set of a matrix and we have showed the necessary and sufficient condition for a matrix to be strongly regular.

Chapter 3

Max-algebraic Eigenvalues and Eigenvectors

3.1 Introduction

In the previous chapter, we have introduced the basic concepts of max-algebra. In this chapter, we will introduce one of the most significant topic and some of its key results in max-algebra. We will start by considering the following problem.

Problem 1. Given $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$, find $x \in \overline{\mathbb{R}}^n$, $x \neq \epsilon$ and $\lambda \in \overline{\mathbb{R}}$ s.t.

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

If we replace the max-algebraic operations with the conventional operations, we know that λ is an eigenvalue and x is an eigenvector of A. Therefore, for the above max-algebraic problem, we will call λ to be a *max-algebraic eigenvalue* and x to be a *max-algebraic eigenvector* of A. In general, we will call Problem 1 *max-algebraic eigenvalue-eigenvector problem* or *max-algebraic eigenproblem*.

Finding eigenvalues and eigenvectors in conventional linear algebra is a well known problem and different methods are developed to solve this problem. For max-algebraic eigenproblem, it had been proved that the problem can be solved efficiently and full solution method in the case of irreducible matrices has been presented in [29], [41] and [63]. A general spectral theorem for reducible matrices can be found in [8] and [38]. In this chapter, we will discuss a solution method for finding eigenvalues and eigenvectors.

Since we are only investigating max-algebra problem in this thesis, then for simplicity we will omit the word "max-algebraic" in all the notation we defined above. Therefore unless stated otherwise, for the rest of this thesis when we use the words eigenvalue, eigenvector and eigenproblem, it immediately implies that we are considering max-algebraic eigenvalue, max-algebraic eigenvector and max-algebraic eigenproblem.

3.2 The Steady State Problem

Let us consider the production system from Example 1.3.1 in Chapter 1 again. Suppose that we have the machines are now working interactively and in stages. In each stage, all machines produce components necessary for the next next stage of some or all other machines simultaneously. We will assume that the manufacturers would want each machine to start a new stage again as soon as possible so that they can produce as many components as possible.

The manufacturers will want to find the times they should start each machine over a period of time, i.e the starting time of each machine at the first stage. We will denote $x_i(k)$ to be the starting time of machine M_i, $i = 1, ..., n$, when it is started for the k^{th} time and we will also denote a_{ij} to be the unit of time for machine M_j to produce components necessary for machine M_i. Therefore the second starting time for machine M_i, $i = 1, ..., n$, will be

$$x_i(2) = \max(a_{i1} + x_1(1), ..., a_{in} + x_n(1)).$$

Similarly the starting time for machine M_i when it is started the $k + 1^{th}$ time will be

$$x_i(k+1) = \max(a_{i1} + x_1(k), ..., a_{in} + x_n(k)). \tag{3.1}$$

If we model the above in max-algebraic notation, we can see that it describes a max-linear system and (3.1) becomes

$$x(k+1) = A \otimes x(k). \tag{3.2}$$

Since every machine is likely to perform the same task repeatedly over a period of time; at each stage a machine will start and finish the process and wait until all components for the next stage are ready. One of the criteria the manufacturers may want to meet when choosing the starting times is that the starting times between two consecutive stages differ by the same constant for every machine. That is for some λ, we have for all $k \in \mathbb{N}$

$$x(k+1) = \lambda \otimes x(k). \tag{3.3}$$

Then (3.2) and (3.3) immediately imply that

$$A \otimes x(k) = \lambda \otimes x(k). \tag{3.4}$$

We will say that the production system reaches a *steady state* if we can find a $x(k)$ such that (3.4) holds.

If we choose the first starting time vector, i.e. $x(1)$, to be an eigenvector of A corresponding to an eigenvalue λ, we will then have

$$\begin{aligned}
x(2) &= A \otimes x(1) &&= \lambda \otimes x(1), \\
x(3) &= A \otimes x(2) &&= A \otimes \lambda \otimes x(1), \\
& &&= \lambda \otimes A \otimes x(1), \\
& &&= \lambda \otimes x(2), \\
\dots \\
x(k) &= A \otimes x(k-1) &&= A \otimes \lambda \otimes x(k-2), \\
& &&= \lambda \otimes A \otimes x(k-2), \\
& &&= \lambda \otimes x(k-1).
\end{aligned}$$

Therefore the production system has reached a steady state immediately if and only if there exists λ and x such that x is a solution to

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

It immediately follows that solving the steady state problem is equivalent to solving the eigenproblem.

3.3 Basic Principles

Before we can present a solution method for solving the eigenproblem, we will need to define some of the necessary notations and definitions for this problem. It turns out that the results for eigenproblem are closely related to graph theory. Therefore we will start by considering some of the definitions in graph theory. First we will define the concept of a directed graph.

Definition 3.3.1. Suppose that we have the sets V, E, where $V \neq \emptyset$ is a finite set of elements (the set of nodes) and $E \subseteq V \times V$ where E contains a set of ordered pairs of nodes (the set of edges or arcs), then $D = (V, E)$ is called a *directed graph* or a *digraph*.

Using the definition of a directed graph, we can obtain the definition of a weighted directed graph.

Definition 3.3.2. Let $D = (V, E, w)$ where (V, E) is a digraph and $w : E \to \overline{\mathbb{R}}$, then D is called a *weighted directed graph* or a *weighted digraph*.

Next we will consider the concepts of path, cycle and elementary cycle in a digraph.

Definition 3.3.3. Suppose that $D = (V, E)$ is a digraph, then $\pi = (v_1, ..., v_{p+1})$ is called a *path* if $(v_1, ..., v_{p+1})$ is a sequence of nodes, i.e.

$$v_i \in V, \ \ \forall i = 1, ..., p+1 \text{ and}$$
$$(v_i, v_{i+1}) \in E \ \ \forall i = 1, ...p.$$

We will say the path π has length p and we will call v_1 the *starting node* and v_{p+1} the *endnode* of π.

Definition 3.3.4. Suppose that $D = (V, E, w)$ is a weighted digraph and $\pi = (v_1, ..., v_{p+1})$ is a path from v_1 to v_{p+1}, then the weight for the path π is equal to

$$w(v_1, v_2) + w(v_2, v_3) + ... + w(v_p, v_{p+1}).$$

Definition 3.3.5. Suppose that $D = (V, E)$ is a digraph, then $\sigma = (v_1, ..., v_{p+1})$ is called a *cycle* if σ is a path and $v_1 = v_{p+1}$. The length of the cycle is said to have length p.

Definition 3.3.6. Suppose that $D = (V, E)$ is a digraph, then $\sigma = (v_1, ..., v_{p+1})$ is called an *elementary cycle* if σ is a cycle and $v_i \neq v_j$ for all $i, j = 1, ..., p$ and $i \neq j$.

Example 3.3.1. (a, d, e) is not a path.
(a, b, c, d, b, e) is a path but not a cycle.

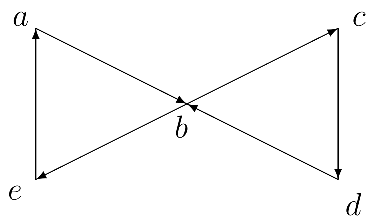

Figure 3.1: Example 3.3.1

(a, b, c, d, b, e, a) is a cycle but not elementary.

(b, c, d, b) is an elementary cycle.

Next we will need to define the notion of strong connectivity in a directed graph.

Definition 3.3.7. Suppose that $D = (V, E)$ is a digraph and $u, v \in V$, then v is said to be *reachable* from u if there exists a path in D from u to v. We will denote this by $u \to v$.

Definition 3.3.8. Suppose that $D = (V, E)$ is a digraph, then D is called a *strongly connected* graph if u is reachable from v for all $u, v \in V$.

We can also for every matrix A, obtain a weighted directed graph associated with A. It is defined by the following:

Definition 3.3.9. Suppose that $A = (a_{ij}) \in \overline{\mathbb{R}}^{n \times n}$, then the *associated weighted digraph* of A is

$$D_A = (N, E = \{(i, j) \mid a_{ij} > \epsilon\}, w)$$

where $w(i, j) = a_{ij}$ for all $(i, j) \in E$.

Using the above definitions, we can now define irreducible and reducible matrices.

Definition 3.3.10. Suppose that $A = (a_{ij}) \in \overline{\mathbb{R}}^{n \times n}$ and D_A is the associated weighted digraph of A, then A is called *irreducible* if D_A is strongly connected. A is called *reducible* otherwise.

Note that A is irreducible if $n = 1$.

Also we will define the notion of metric matrices.

Definition 3.3.11. Suppose that $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$, then we will define the following infinite series

$$\Gamma(A) = A \oplus A^2 \oplus ....$$

If these series converge to matrices that do not contain $+\infty$, then we will call $\Gamma(A)$ the *metric matrix* and sometimes it is also called the *weak transitive closure* of A.

Note that if we let D_A be the associated weighted digraph of A and the matrix $A^2 = A \otimes A$, we can see that for each element in A^2 we have

$$\sum_{k=1,...,n}^{\oplus} a_{ik} \otimes a_{kj} = \max(a_{ik} + a_{kj}).$$

This is equivalent to the weight of the heaviest $i - j$ paths of length 2 in D_A and similarly for the elements of A^k, $k = 1, 2, ...$ Therefore the matrix $\Gamma(A)$ represents the weights of the heaviest paths of any length for all pairs of nodes in D_A.

Definition 3.3.12. Suppose that $A \in \overline{\mathbb{R}}^{n\times n}$ and D_A is the associated weighted digraph of A. Let σ be a cycle in D_A, then

$$\mu(\sigma, A) = \frac{w(\sigma, A)}{l(\sigma)}$$

where $w(\sigma)$ is the weight of the cycle and $l(\sigma)$ is the length of the cycle. We will call $\mu(\sigma, A)$ the *cycle mean* of cycle σ with respect to matrix A. We will also let

$$\lambda(A) = \max_{\sigma} \ \mu(\sigma, A)$$

and we will call $\lambda(A)$ the *maximum cycle mean* of A. Since $\max \emptyset = \epsilon$, it immediately follows from the definition that D_A is acyclic if and only if $\lambda(A) = \epsilon$.

Note that a lot of different algorithms are developed for finding maximum cycle mean. One of the fastest and most commonly used algorithm is Karp's algorithm [44] with computational complexity $O(mn)$ where m is the number of arcs in D_A.

Theorem 3.3.1. [30] Suppose $A \in \overline{\mathbb{R}}^{n\times n}$ and $\lambda(A) > \epsilon$, then $\forall \alpha \in \mathbb{R}$,

$$\lambda(\alpha \otimes A) = \alpha \otimes \lambda(A).$$

It is not difficult to prove that the associated weighted digraph of any irreducible matrix contains at least one cycle, therefore the maximum cycle mean for any irreducible matrix is finite. Now we will need to consider some of the properties of matrices when their maximum cycle mean is equal to 0.

Definition 3.3.13. Suppose that $A \in \overline{\mathbb{R}}^{n\times n}$, then A is called *definite* if $\lambda(A) = 0$.

Note that by using Proposition 3.3.1, we can see that for any irreducible matrix, we can always generate the matrix $A_\lambda = \lambda(A)^{-1} \otimes A$ such that A_λ is definite. For simplicity, we will denote $A_\lambda = \lambda(A)^{-1} \otimes A$ from this point forward.

Theorem 3.3.2. [29] Suppose that $A \in \overline{\mathbb{R}}^{n\times n}$ is definite, then

$$\Gamma(A) = A \oplus A^{(2)} \oplus ... \oplus A^{(n)}.$$

3.4 Principle Eigenvalue

Now we will discuss a solution method on finding all eigenvalues for any matrix. Throughout this chapter and the rest of this thesis, we will use the following notation:

Definition 3.4.1. Let $A \in \overline{\mathbb{R}}^{n\times n}$ and $\lambda \in \overline{\mathbb{R}}$. Let us define

$$\begin{aligned} V(A,\lambda) &= \{x \in \overline{\mathbb{R}}^n \mid A \otimes x = \lambda \otimes x\}, \\ \Lambda(A) &= \{\lambda \in \overline{\mathbb{R}} \mid V(A,\lambda) \neq \{\epsilon\}\}, \\ V(A) &= \bigcup_{\lambda\in\Lambda(A)} V(A,\lambda), \\ V^+(A,\lambda) &= V(A,\lambda) \cap \mathbb{R}^n, \\ V^+(A) &= V(A) \cap \mathbb{R}^n. \end{aligned}$$

In other words, $V(A,\lambda)$ represents ϵ and the set of eigenvectors of A corresponding to the value λ, $\Lambda(A)$ represents the set of all possible eigenvalues of A and $V(A)$ represents the set of all eigenvectors of A and ϵ. Also we have $V^+(A,\lambda)$ represents the set of finite eigenvectors of A corresponding to the value λ and $V^+(A)$ represents the set of all finite eigenvectors of A.

We will first present some of the basic properties of eigenvalues and eigenvectors.

Proposition 3.4.1. Let A, $B \in \overline{\mathbb{R}}$, $\alpha \in \mathbb{R}$, λ, $\mu \in \overline{\mathbb{R}}$ and x, $y \in \overline{\mathbb{R}}^n$. Then

1. $V(\alpha \otimes A) = V(A)$.
2. $\Lambda(\alpha \otimes A) = \alpha \otimes \Lambda(A)$.
3. $V(A,\lambda) \cap V(B,\mu) \subseteq V(A \otimes B, \lambda \otimes \mu)$.
4. $V(A,\lambda) \cap V(B,\mu) \subseteq V(A \oplus B, \lambda \oplus \mu)$.

Proof. If $A \otimes x = \lambda \otimes x$ then it immediately follows that $(\alpha \otimes A) \otimes x = (\alpha \otimes \lambda) \otimes x$. This proves 1 and 2. Let us suppose that $A \otimes x = \lambda \otimes x$ and $B \otimes x = \mu \otimes x$, then

$$\begin{aligned}(A \otimes B) \otimes x &= A \otimes (B \otimes x),\\ &= A \otimes \mu \otimes x,\\ &= \mu \otimes A \otimes x,\\ &= \mu \otimes \lambda \otimes x\end{aligned}$$

which proves 3. Also we know that

$$\begin{aligned}(A \otimes B) \otimes x &= A \otimes x \oplus B \otimes x,\\ &= \lambda \otimes x \oplus \mu \otimes x,\\ &= (\lambda \oplus \mu) \otimes x,\end{aligned}$$

which proves 4. □

It turns out that the concept of maximum cycle mean have a significant property when considering the eigenproblem. From the following part of this section, we will show how the maximum cycle mean of a matrix have played an important role when solving the eigenproblem.

Definition 3.4.2. Suppose that $A \in \overline{\mathbb{R}}^{n\times n}$, then we will denote

$$E(A) = \{i \in N \mid \exists \sigma = (i = i_1, i_2, ..., i_k, i_1) \text{ in } D_A \text{ s.t. } \mu(\sigma, A) = \lambda(A)\}.$$

The elements of $E(A)$ are called *eigen-nodes* or *critical nodes* of A. A cycle σ is called a *critical cycle* if $\mu(\sigma, A) = \lambda(A)$.

By using the set of nodes N; the union of the sets of arcs of all critical cycles, we can generate a digraph $C(A)$ and we will call $C(A)$ the *critical digraph* of A.

Lemma 3.4.1. [6][20] Let $A \in \overline{\mathbb{R}}^{n\times n}$ and $C(A)$ be the critical digraph of A, then all cycles in $C(A)$ are critical cycles.

We will say that two nodes i and j in $C(A)$ are *equivalent* if i and j belong to the same critical cycle of A. We will denote this relation by the notation $i \sim j$. Note that it is not difficult to prove that $\sim$ constitutes an equivalence relation in $E(A)$.

Lemma 3.4.2. [29] Let $A \in \overline{\mathbb{R}}^{n \times n}$. If $\lambda(A) = \epsilon$, then $\Lambda(A) = \{\epsilon\}$ and the eigenvectors of A are exactly the vectors $(x_1, ..., x_n)^T \in \overline{\mathbb{R}}^n$ such that $x_j = \epsilon$ whenever the j^{th} column of A is not equal to ϵ, $j \in N$.

The above lemma solves the case when $\lambda(A) = \epsilon$, therefore we will usually assume that $\lambda(A) > \epsilon$.

Theorem 3.4.3. [3][29] Suppose $A \in \overline{\mathbb{R}}^{n \times n}$ then $\lambda(A)$ is an eigenvalue for the matrix A. Furthermore if $\lambda(A) > \epsilon$, then up to n eigenvectors corresponding to $\lambda(A)$ can be found among the columns of $\Gamma(A_\lambda)$. More precisely every column of $\Gamma(A_\lambda)$ with zero diagonal entry is an eigenvector of A with corresponding eigenvalue $\lambda(A)$ and we obtain a basis of $V(A, \lambda(A))$ by taking exactly one g_k for each equivalence class in $(E(A), \sim)$.

Example 3.4.1. Consider the matrix

$$A = \begin{pmatrix} 3 & 8 & 0 & 3 & 1 & 7 \\ 4 & 2 & 1 & 3 & 2 & 5 \\ 2 & 3 & 0 & 1 & 4 & 2 \\ 2 & 1 & 3 & 6 & 5 & 1 \\ 3 & 0 & 1 & 2 & 3 & 5 \\ 0 & 1 & 1 & 8 & 2 & 3 \end{pmatrix}.$$

Using Karp's algorithm, we can deduce that the maximum cycle mean is 6 and it is attained by three critical cycles; $(1,2,1)$, $(4,4)$ and $(4,5,6,4)$. Therefore $\lambda(A)=6$ and we have

$$\Gamma(A_\lambda)=\begin{pmatrix} 0 & 2 & 0 & 3 & 2 & 1 \\ -2 & 0 & -2 & 1 & 0 & -1 \\ -4 & -2 & -4 & -1 & -2 & -3 \\ -4 & -2 & -3 & 0 & -1 & -2 \\ -3 & -1 & -2 & 1 & 0 & -1 \\ -2 & 0 & -1 & 2 & 1 & 0 \end{pmatrix}.$$

We can see that the critical digraph $C(A)$ has two strongly connected components; one with the node set $\{1,2\}$ and the other one with the node set $\{4,5,6\}$. Therefore there are two equivalence classes in $(E(A),\sim)$, hence the first and second column of $\Gamma(A_\lambda)$ are multiples of each other and similarly for the fourth, fifth and sixth columns. For the basis of $V(A,\lambda(A))$ we may take the first and fourth column.

Example 3.4.2. Consider the matrix

$$A=\begin{pmatrix} 0 & 3 & \epsilon & \epsilon \\ 5 & 1 & \epsilon & \epsilon \\ \epsilon & 3 & 0 & \epsilon \\ 2 & 1 & \epsilon & 4 \end{pmatrix}.$$

Then $\lambda(A)=4$, $E(A)=\{1,2,4\}$ and

$$\Gamma(A_\lambda)=\begin{pmatrix} 0 & -1 & \epsilon & \epsilon \\ 1 & 0 & \epsilon & \epsilon \\ 0 & -1 & -4 & \epsilon \\ -2 & -3 & \epsilon & 0 \end{pmatrix}.$$

Therefore the basis of $V(A, \lambda(A))$ is $\{(0, 1, 0, -2)^T, (\epsilon, \epsilon, \epsilon, 0)^T\}$.

From Theorem 3.4.3, we can see that $\lambda(A)$ is of a special significance as an eigenvalue; it is an eigenvalue for every matrix. It will follow from the Spectral Theorem (Theorem 3.5.4) that $\lambda(A)$ is the greatest eigenvalue. We will therefore call $\lambda(A)$ the principal eigenvalue of A and the subspace $V(A, \lambda(A))$ will be called the principal eigenspace of A.

Now we will show that $\lambda(A)$ is the only eigenvalue whose corresponding eigenvectors may be finite.

Theorem 3.4.4. [29] Suppose $A \in \overline{\mathbb{R}}^{n\times n}$, $A \neq \epsilon$ and $V^+(A) \neq \emptyset$, then $\lambda(A) > \epsilon$ and $A \otimes x = \lambda(A) \otimes x, \forall x \in V^+(A)$.

Note that if $A = \epsilon$ then every finite vector of a suitable dimension is an eigenvector of A and all correspond to the eigenvalue $\lambda(A) = \epsilon$.

Next we will present one classical result in max-algebra.

Theorem 3.4.5. [29] Suppose $A \in \overline{\mathbb{R}}^{n\times n}$ and $A \neq \epsilon$, then the following hold:

1. $V^+(A) \neq \emptyset \iff \lambda(A) > \epsilon$ and in D_A, $\forall j \in N$,

$$\exists i \in E(A) \text{ such that } j \to i.$$

2. If, moreover, $V^+(A) \neq \emptyset$ then

$$V^+(A) = \{\sum_{j\in E(A)}^{\oplus} \alpha_j \otimes g_j;\ \alpha_j \in \mathbb{R}\}$$

where $g_1, ..., g_n$ are the columns of $\Gamma(A_\lambda)$.

The above theorem shows us the necessary and sufficient conditions for the existence of finite eigenvectors. It also shows us how to generate the set of finite eigenvectors. Using

Theorem 3.4.5 and the following result, we are able to device a slightly more efficient way on generating the set of finite eigenvectors.

Theorem 3.4.6. [29] Suppose $A \in \overline{\mathbb{R}}^{n\times n}$ is definite, $\lambda(A) > \epsilon$, $\Gamma(A) = (g_{ij})$ and $g_1, ..., g_n$ are the columns of $\Gamma(A)$. Then

- $i \in E(A) \Longleftrightarrow g_{ii} = 0.$
- If $i, j \in E(A)$ then $g_i = \alpha \otimes g_j$ for some $\alpha \in \mathbb{R}$ if and only if $i \sim j$.

Corollary 3.4.1. Suppose $A \in \overline{\mathbb{R}}^{n\times n}$. If $\lambda(A) > \epsilon$, $g_1, ..., g_n$ are the columns of $\Gamma(A_\lambda)$ and $V^+(A) \neq \emptyset$, then

$$V^+(A) = \{ \sum_{j\in E^*(A)}^{\oplus} \alpha_j \otimes g_j;\ \alpha_j \in \mathbb{R}\}$$

where $E^*(A)$ is any maximal set of non-equivalent critical nodes of A. The size of $E^*(A)$ is equal to the number of non-trivial strongly connected components of the critical digraph $C(A)$.

Using the results we obtained above, we can now deduce the following classical complete solution of the eigenproblem for irreducible matrices.

Theorem 3.4.7. [29] Every irreducible matrix $A \in \overline{\mathbb{R}}^{n\times n}$, $(n > 1)$ has a unique eigenvalue equal to $\lambda(A)$ and

$$V(A) - \{\epsilon\} = V^+(A) = \{ \sum_{j\in E*(A)}^{\oplus} \alpha_j \otimes g_j;\ \alpha_j \in \mathbb{R}\}$$

where $g_1, ..., g_n$ are the columns of $\Gamma(A_\lambda)$ and $E^*(A)$ is any maximal set of non-equivalent critical nodes of A.

Note that the 1×1 matrix $A = (\epsilon)$ is irreducible and $V(A) = V^+(A) = \mathbb{R}$.

If $n > 1$, by the definition of irreducible matrices, we know that $\forall i, j \in N$, $i \rightarrow j$ and therefore the weight of the heaviest $i - j$ path of any length is not equal to ϵ. Hence $\Gamma(A_\lambda)$ for an irreducible matrix ($n > 1$) A is finite.

We should also note that for an irreducible matrix A

$$V(A) = V^+(A) \cup \{\epsilon\} = \{\Gamma(A_\lambda) \otimes z; z \in \overline{\mathbb{R}}^n, z_j = \epsilon, \forall j \notin E(A)\}.$$

The fact that $\lambda(A)$ is the unique eigenvalue of an irreducible matrix A was proved in [27] and then independently in [63] for finite matrices.

Example 3.4.3. Consider the irreducible matrix

$$A = \begin{pmatrix} 0 & 3 & 0 & 1 \\ 5 & 1 & \epsilon & \epsilon \\ \epsilon & 3 & 0 & \epsilon \\ 2 & 1 & \epsilon & 4 \end{pmatrix}.$$

Then $\lambda(A) = 4$, $E(A) = \{1, 2, 4\}$ and

$$\Gamma(A_\lambda) = \begin{pmatrix} 0 & -1 & -4 & -3 \\ 1 & 0 & -3 & -2 \\ 0 & -1 & -4 & -3 \\ -2 & -3 & -6 & 0 \end{pmatrix}.$$

Hence the basis of the principal eigenspace is $\{(0, 1, 0, -2)^T, (-3, -2, -3, 0)^T\}$.

3.5 Finding All Eigenvalues

Before we produce a method for finding all eigenvalues of a matrix, we need to introduce some notation that will be useful.

Definition 3.5.1. If $1 \leq i_1 < i_2 < ... < i_k \leq n$, $K = \{i_1, ..., i_k\} \subseteq N$, then $A[K]$ denotes the principal submatrix

$$\begin{pmatrix} a_{i_1 i_1} & ... & a_{i_1 i_k} \\ ... & ... & ... \\ a_{i_k i_1} & ... & a_{i_k i_k} \end{pmatrix}$$

of the matrix $A = (a_{ij})$ and $x[K]$ denotes the subvector $(x_{i_1}, ..., x_{i_k})^T$ of the vector $x = (x_1, ..., x_n)^T \in \overline{\mathbb{R}}^n$. Furthermore, if $D = (N, E)$ is a digraph and $K \subseteq N$ then $D[K]$ denotes the *induced subgraph* of D; that is

$$D[K] = (K, E \cap (K \times K)).$$

It is not difficult to see that $D_{A[K]} = D[K]$.

Definition 3.5.2. Suppose $A, B \in \overline{\mathbb{R}}^{n\times n}$, then the symbol $A \sim B$ means that A can be obtained from B by a simultaneous permutation of rows and columns.

We can see that if $A \sim B$, then the induced digraph D_A can be obtained from D_B by a renumbering of the nodes. Hence if $A \sim B$ then A is irreducible if and only if B is irreducible.

Lemma 3.5.1. [21] If $A \sim B$ then $\Lambda(A) = \Lambda(B)$ and there is a bijection between $V(A)$ and $V(B)$.

The following lemma gives a clear signal that also in max-algebra the Frobenius normal form will be useful for describing all eigenvalues.

Lemma 3.5.2. [21] Suppose $A \in \overline{\mathbb{R}}^{n\times n}$, $\lambda \in \Lambda(A)$ and $x \in V(A,\lambda)$. If $x \notin V^+(A,\lambda)$, then $n > 1$,

$$A \sim \begin{pmatrix} A^{(11)} & \epsilon \\ A^{(21)} & A^{(22)} \end{pmatrix},$$

$\lambda = \lambda(A^{(22)})$ and hence A is reducible.

Proposition 3.5.1. [21] Let $A \in \overline{\mathbb{R}}^{n\times n}$, then $V(A) = V^+(A)$ if and only if A is irreducible.

Every matrix $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$ can be transformed in linear time by simultaneous permutations of the rows and columns to a *Frobenius normal form* (FNF) [57], i.e.

$$\begin{pmatrix} A_{11} & \epsilon & ... & \epsilon \\ A_{21} & A_{22} & ... & \epsilon \\ ... & ... & ... & ... \\ A_{r1} & A_{r2} & ... & A_{rr} \end{pmatrix} \tag{3.5}$$

where $A_{11}, ..., A_{rr}$ are irreducible square submatrices of A.

If A is in an FNF then we will denote the sets $N_1, ..., N_r$ to be the corresponding partition of the node set N of D_A. We will call these sets classes (of A). Since all square submatrix $A_{11}, ..., A_{rr}$ are irreducible, it follows that each of the induced subgraph $D_A[N_i]$, $(i = 1, ..., r)$ is strongly connected and a path from N_i to N_j in D_A exists only if $i \to j$. We will also say for simplicity that $\lambda(A_{jj})$ is the eigenvalue of the class N_j.

Definition 3.5.3. Let A be in an FNF, then the *condensation digraph* is the digraph

$$C_A = (\{N_1, ..., N_r\}, \{(N_i, N_j) \mid \exists k \in N_i,\ \exists l \in N_j \text{ s.t. } a_{kl} > \epsilon\}).$$

We will denote the symbol $N_i \to N_j$ if there is a directed path from a node in N_i to a node in N_j in D_A. Therefore if $N_i \to N_j$, there exists a directed path from each node in N_i to each node in N_j. Equivalently, there is a directed path from N_i to N_j in C_A.

If there are neither outgoing nor incoming arcs from or to an induced subgraph

$$C_A[\{N_{i_1}, ..., N_{i_s}\}],\ (1 \le i_1 < ... < i_s \le r)$$

and no proper subdigraph has this property then the submatrix

$$\begin{pmatrix} A_{i_1 i_1} & \epsilon & ... & \epsilon \\ A_{i_2 i_1} & A_{i_2 i_2} & ... & \epsilon \\ ... & ... & ... & ... \\ A_{i_s i_1} & A_{i_s i_2} & ... & A_{i_s i_s} \end{pmatrix}$$

is called an *isolated superblock* or just *superblock*. The induced subdigraph of C_A corresponding to an isolated superblock is a directed tree (although the underlying undirected graph is not necessarily acyclic). C_A is the union of a number of such directed trees. The nodes of C_A with no incoming arcs are called the *initial classes*, those with no outgoing arcs are called the *final classes*. Note that the directed tree corresponding to an isolated superblock may have several initial and final classes.

For instance the condensation digraph for the matrix

$$\begin{pmatrix} A_{11} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ * & A_{22} & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & * & A_{33} & \epsilon & \epsilon & \epsilon \\ * & * & \epsilon & A_{44} & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & A_{55} & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & * & A_{66} \end{pmatrix} \tag{3.6}$$

can be seen in Figure 3.2 (note that here and elsewhere the symbols $*$ indicate submatrices different from ϵ). It consists of six classes and two superblocks; $\{N_1, N_2, N_3, N_4\}$ and

$\{N_5, N_6\}$. The classes N_3, N_4 and N_6 are initial and N_1 and N_5 are final classes.

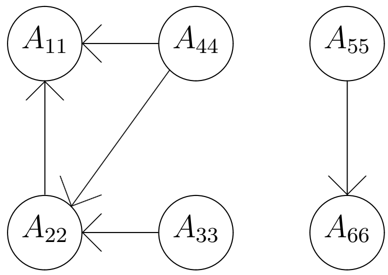

Figure 3.2: Condensation digraph for matrix (3.6)

Lemma 3.5.3. [21] If $x \in V(A)$, $N_i \to N_j$ and $x[N_j] \neq \epsilon$ then $x[N_i]$ is finite. In particular, $x[N_j]$ is finite.

The following key result has appeared in the thesis [38] and [7]. The latter work refers to the report [8] for a proof.

Theorem 3.5.4. [38](Spectral Theorem) Let $A \in \overline{\mathbb{R}}^{n\times n}$ be in the FNF. Then

$$\Lambda(A) = \{\lambda(A_{jj}) \mid \lambda(A_{jj}) = \max_{N_i \to N_j} \lambda(A_{ii})\}.$$

Note that significant correlation exists between the max-algebraic spectral theory and that for non-negative matrices in linear algebra [9],[57], [61]. For instance the Frobenius normal form and accessibility between classes are essentially the same. The maximum cycle mean corresponds to the Perron root for irreducible (nonnegative) matrices and finite eigenvectors in max-algebra correspond to positive eigenvectors in the non-negative spectral theory.

Definition 3.5.4. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be in the FNF. If

$$\lambda(A_{jj}) = \max_{N_i \to N_j} \lambda(A_{ii})$$

then A_{jj} (and also N_j or just j) will be called *spectral*.

Thus $\lambda(A_{jj}) \in \Lambda(A)$ if j is spectral but not necessarily the other way round. The following corollaries of the spectral theorem are readily proved.

Corollary 3.5.1. [21] All initial classes of C_A are spectral.

Corollary 3.5.2. [21] $1 \leq |\Lambda(A)| \leq n$ for every $A \in \overline{\mathbb{R}}^{n\times n}$.

Corollary 3.5.3. [21] $V(A) = V(A, \lambda(A))$ if and only if all initial classes have the same eigenvalue $\lambda(A)$.

Example 3.5.1. Let us consider the condensation digraph in Figure 3.3. It contains 10 classes including two initial classes and four final classes. The integers indicate the eigenvalues of the corresponding classes. The number in bold indicate the corresponding class is spectral, the others are not.

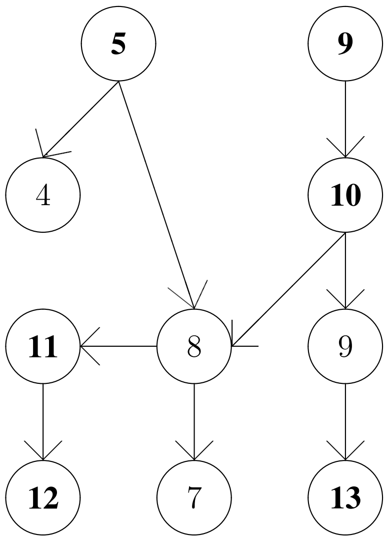

Figure 3.3: Condensation digraph

3.6 Finding All Eigenvectors

Note that the unique eigenvalue of every class; that is of a diagonal block of an FNF, can be found in $O(n^3)$ time by applying Karp's algorithm to each block. The condition for identifying all spectral submatrices in an FNF provided in Theorem 3.5.4 enables us to find them in $O(r^2) \leq O(n^2)$ time by applying standard reachability algorithms to C_A.

Definition 3.6.1. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be in the FNF, $N_1, ..., N_r$ be the classes of A and $R = \{1, ..., r\}$. Suppose that $\lambda \in \Lambda(A)$ and $\lambda > \epsilon$, then we will denote

$$I(\lambda) = \{i \in R \mid \lambda(N_i) = \lambda,\ N_i \text{ spectral}\}$$

and

$$E(\lambda) = \bigcup_{i\in I(\lambda)} E(A_{ii}) = \{j \in N \mid g_{jj} = 0,\ j \in \bigcup_{i\in I(\lambda)} N_i\}.$$

where $\Gamma(\lambda^{-1} \otimes A) = (g_{ij})$.

Note that $\Gamma(\lambda^{-1} \otimes A)$ may now include entries equal to $+\infty$.

Definition 3.6.2. Let $i, j \in E(\lambda)$, then the nodes i and j are called λ-*equivalent* if i and j belong to the same cycle of cycle mean λ. We will denote this by $i \sim_\lambda j$.

Theorem 3.6.1. [21] Suppose $A \in \overline{\mathbb{R}}^{n\times n}$ and $\lambda \in \Lambda(A)$, $\lambda > \epsilon$. Then $g_j \in \overline{\mathbb{R}}^n$ for all $j \in E(\lambda)$ and a basis of $V(A, \lambda)$ can be obtained by taking one g_j for each $\sim_\lambda$ equivalence class.

Corollary 3.6.1. [21] A basis of $V(A, \lambda)$ for $\lambda \in \Lambda(A)$ can be found using $O(n^3)$ operations and we have

$$V(A, \lambda) = \{\Gamma(\lambda^{-1} \otimes A) \otimes z \mid z \in \overline{\mathbb{R}}^n, z_j = \epsilon \text{ for all } j \notin E(\lambda)\}.$$

Theorem 3.6.2. [21] $V^+(A) \neq \emptyset$ if and only if $\lambda(A)$ is the eigenvalue of all final classes.

Corollary 3.6.2. [21] $V^+(A) \neq \emptyset$ if and only if a final class has eigenvalue less than $\lambda(A)$.

Note that a final class with eigenvalue less than $\lambda(A)$ may not be spectral and so $\Lambda(A) = \{\lambda(A)\}$ is possible even if $V^+(A) = \emptyset$. For instance in the case of

$$A = \begin{pmatrix} 1 & \epsilon & \epsilon \\ \epsilon & 0 & \epsilon \\ 0 & 0 & 1 \end{pmatrix}.$$

We have $\lambda(A) = 1$, but $V^+(A) = \emptyset$.

3.7 Formulation of the Problem

Using the results we have discussed previously, we are able to describe the set of eigenvector for each corresponding eigenvalue as an image set of a matrix by the following proposition.

Proposition 3.7.1. Suppose that $A \in \overline{\mathbb{R}}^{n\times n}$ be in the FNF and $N_1, ..., N_r$ be the classes of A, $R = \{1, ..., r\}$. Let $K = \{1, ..., k\}$ and $\Lambda(A) = \{\lambda_i \mid i \in K\}$, then $k \le r$ and we can obtain matrices $\Gamma^{(1)}, ..., \Gamma^{(k)}$ such that $\forall i \in K$,

$$Im(\Gamma^{(i)}) = V(A, \lambda_i).$$

Proof. Since the number of eigenvalues cannot exceed the number of classes, it immediately follows that $k \le r$. Let $\lambda_i \in \Lambda(A)$ then by Corollary 3.6.1

$$V(A, \lambda_i) = \{\Gamma(\lambda_i^{-1} \otimes A) \otimes z \mid z \in \overline{\mathbb{R}}^n, z_j = \epsilon \text{ for all } j \notin E(\lambda_i)\}.$$

Suppose that $E(\lambda_i) = \{e_1, ..., e_l\}$, then if we let $\Gamma^{(i)} = (g_{ij})$ and $g_1, ..., g_l$ be the columns of $\Gamma^{(i)}$ where g_i is equal to the e_i^{th} column of $\Gamma(\lambda_i^{-1} \otimes A)$ then we have

$$Im(\Gamma^{(i)}) = V(A, \lambda_i).$$

□

Example 3.7.1. Consider the following matrix

$$A = \left(\begin{array}{c|cc|cc|ccc} 3 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \hline \epsilon & -1 & 0 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 1 & 2 & 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \hline \epsilon & 2 & \epsilon & 2 & 1 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 0 & 2 & \epsilon & \epsilon & \epsilon \\ \hline \epsilon & 1 & \epsilon & \epsilon & \epsilon & 1 & 4 & 2 \\ 3 & \epsilon & \epsilon & \epsilon & \epsilon & 2 & 2 & 1 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 0 & -1 & 3 \end{array}\right). \tag{3.7}$$

Since A is in the FNF, then we can see that

$$\begin{aligned} A_{11} &= (3), & A_{22} &= \begin{pmatrix} -1 & 0 \\ 2 & 2 \end{pmatrix} \\ A_{33} &= \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}, & A_{44} &= \begin{pmatrix} 1 & 4 & 2 \\ 2 & 2 & 1 \\ 0 & -1 & 3 \end{pmatrix} \end{aligned}$$

with $\lambda(A_{11}) = 3$, $\lambda(A_{22}) = 2$, $\lambda(A_{33}) = 2$ and $\lambda(A_{44}) = 3$.

Since N_3 and N_4 are initial blocks, then it immediately implies that they are spectral. We can also see that N_1 is spectral since

$$\lambda(A_{11}) = \max_{i=1,\dots,4} \lambda(A_{ii}).$$

N_2 is not a spectral block because N_2 is reachable from N_4 and $\lambda(A_{44}) > \lambda(A_{22})$. Therefore we have $\Lambda(A) = \{2, 3\}$. Let us denote $\lambda_1 = 2$ and $\lambda_2 = 3$.

First we will consider when $\lambda = \lambda_1$. We will have $I(\lambda_1) = \{3\}$ and $E(\lambda_1) = \{4, 5\}$. Next we will calculate $\Gamma(\lambda_1^{-1} \otimes A)$ and we will obtain the following matrix:

$$\begin{pmatrix} 8 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 3 & -2 & -2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 6 & 0 & 0 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 2 & 0 & -2 & 0 & -1 & \epsilon & \epsilon & \epsilon \\ -1 & -2 & -4 & -2 & 0 & \epsilon & \epsilon & \epsilon \\ 9 & 5 & 3 & \epsilon & \epsilon & 8 & 8 & 7 \\ 8 & 5 & 1 & \epsilon & \epsilon & 6 & 8 & 6 \\ 6 & 3 & 0 & \epsilon & \epsilon & 5 & 6 & 8 \end{pmatrix}$$

From the matrix $\Gamma(\lambda_1^{-1} \otimes A)$, we can obtain $\Gamma^{(1)} = (g_1^{(1)}, g_2^{(1)})$ where $g_1^{(1)}$ equals to the 4^{th} column and $g_2^{(1)}$ equals to the 5^{th} column of $\Gamma(\lambda_1^{-1} \otimes A)$. Therefore

$$\Gamma^{(1)} = \begin{pmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ 0 & -1 \\ -2 & 0 \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \end{pmatrix}.$$

Next we will consider the case when $\lambda = \lambda_2$. We will have $I(\lambda_2) = \{1, 4\}$ and $E(\lambda_2) =$

$\{1, 6, 7, 8\}$. Again we will calculate $\Gamma(\lambda_2^{-1} \otimes A)$ and we will obtain the following matrix:

$$\begin{pmatrix} 0 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ -5 & -4 & -3 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ -2 & -1 & -1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ -6 & -1 & -4 & -1 & -2 & \epsilon & \epsilon & \epsilon \\ -9 & -4 & -7 & -3 & -1 & \epsilon & \epsilon & \epsilon \\ 1 & -2 & -5 & \epsilon & \epsilon & 0 & 1 & -1 \\ 0 & -3 & -6 & \epsilon & \epsilon & -1 & 0 & -2 \\ -2 & -5 & -8 & \epsilon & \epsilon & -3 & -2 & 0 \end{pmatrix}$$

Using the above matrix, we can obtain

$$\Gamma^{(2)} = \begin{pmatrix} 0 & \epsilon & \epsilon \\ -5 & \epsilon & \epsilon \\ -2 & \epsilon & \epsilon \\ -6 & \epsilon & \epsilon \\ -9 & \epsilon & \epsilon \\ 1 & 0 & -1 \\ 0 & -1 & -2 \\ -2 & -3 & 0 \end{pmatrix}$$

Using Proposition 3.7.1, the problem of finding eigenvectors with specified properties have been converted to the problem of finding an image of the mapping

$$x \longrightarrow \Gamma^{(i)} \otimes x, \; i = 1, 2, ...,$$

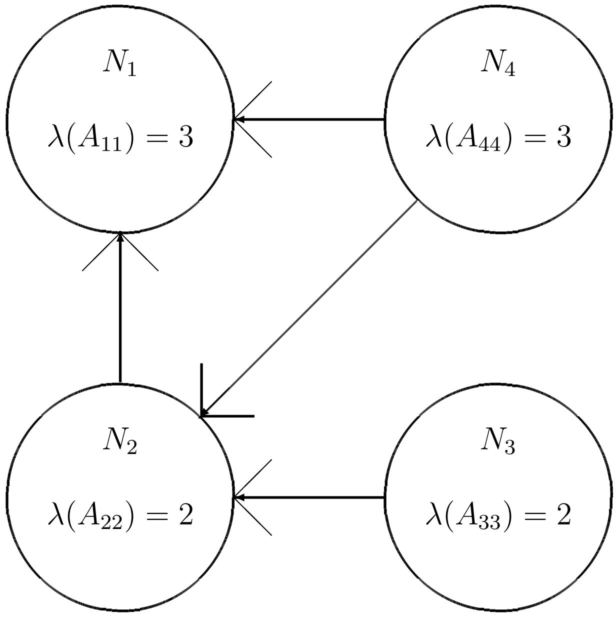

Figure 3.4: Condensation digraph for matrix (3.7)

i.e. the image set of $\Gamma^{(i)}$, with such properties.

Using the result we obtained we above, we will for the rest of this thesis, assume that the matrices $\Gamma^{(i)}$, $i = 1, 2, ...$ are already found. Using these matrices, we will investigate and develop methods for finding eigenvectors for each eigenvalue that may be required by a manufacturer. For simplicity, we will also rename the matrix $\Gamma^{(i)}$ to A from this point forward.

3.8 Summary

In this chapter, we have discussed the steady state problem and how it is related to the max-algebraic eigenproblem. We have seen that this problem is closely related to graph theory. By using this relation, we have presented a solution method for finding all eigenvalues for a matrix.

We have seen that the maximum cycle mean of a matrix plays an important role on solving the eigenproblem. In fact, we now know that the maximum cycle will always be an eigenvalue of any square matrix and it is called the principal eigenvalue. We have shown that the maximum cycle mean is the only eigenvalue such that the corresponding eigenvectors may be finite.

Then we have presented results on finding all the eigenvalues for any square matrices. Using this result, we have seen that we can have at most n distinct eigenvalues where n is the size of the matrix. We have also presented results on finding the set of eigenvectors for each eigenvalue. By this, we can generate a maximum of n rectangular matrices; one for each eigenvalue, such that the set of eigenvectors can be obtained by considering the image set of these matrices. Hence the problem of optimizing eigenvectors has been transformed into optimizing the image set of a matrix.

Chapter 4

Optimizing Range Norm of the Image Set

4.1 Introduction

From Chapter 3, we have transformed the eigenproblem into a linear system problem. Henceforth the aim of this thesis is for a given $A \in \overline{\mathbb{R}}^{m\times n}$, find $b \in \overline{\mathbb{R}}^{m}$ such that $b \in Im(A)$ and b has one of the following property:

1. The difference between the largest and the smallest element in b is minimized/maximized.
2. The difference between the largest and the smallest element in b is minimized/maximized with the condition that some elements in b are already determined.
3. All elements in b are integer, i.e. b is an integer vector.
4. The vector b can be permuted into a specified structure given by the manufacturer.

In this and the following chapter we will investigate the range norm of vectors which are in the image set of any given matrices. The range norm of an vector is the difference between the largest element and the smallest element in the vector.

In real-life situations, there are a lot of different criteria for manufacturers to decide a starting time for their machines. One of them may be to choose the starting time as an eigenvector; so the system will achieve steady state immediately.

Typically it is always the case that there will be more than one independent eigenvector for the manufacturers to choose from. Therefore they may wish to distinguish the one which will suit best to their individual situations inside their factories/plants. So an additional criterion may be required.

One of these additional criteria may be to consider the difference between the earliest and the latest starting time of their machines in the starting time vector. We will denote this difference to be the *range norm* of the starting time vector.

Note that the results in this chapter are not only restricted on optimizing the range norm of eigenvectors; this is merely a special case for these results. In general, optimization of the range norm in an image set of a max-linear system can be thought of optimizing the range norm of the finishing/completion time of a manufacturing process. We shall also note that similar problems were studied in the past; they can be found in [29] and [18] and range norm was called the *range seminorm*.

4.2 Minimizing the Range Norm

When optimizing the starting time, the manufacturers may want all their machines to start at the same time or at least as close as possible. One of the reason for this preference would be that the manufacturers may want to keep track on which stage are the machines working in. This can be easily done when all machines are started at the same time. There may also be a time factor involved with the products after every cycle; the finished products/components may need to ship out at the same time or all the raw materials arrived at the same time and the quality of the materials may deteriorate over time.

Let us start by introducing some definitions and basic results.

Definition 4.2.1. Let $x \in \mathbb{R}^m$, then we will denote the function

$$\delta(x) = \sum_{i \in M}^{\oplus} x_i \; - \sum_{i \in M}^{\oplus'} x_i$$

and call it the *range norm* of x, i.e. range norm = largest value in x - smallest value in x.

Now using the above definition, the problem of minimizing the range norm in an image set can be formulated as below.

Problem 2. Given a matrix $A \in \overline{\mathbb{R}}^{m \times n}$, solve

$$\begin{aligned} \delta(b) \rightarrow \quad & \min \\ \text{subject to} \quad & b \in Im(A). \end{aligned}$$

There may be an instance such that the image set of A contains no finite vectors; this is most likely the case when we consider the fundamental eigenvectors of a reducible matrix. Therefore we will introduce the following modified definition.

Definition 4.2.2. Let $x \in \overline{\mathbb{R}}^m$ then the function

$$\widetilde{\delta}(x) = \sum_{\substack{i \in M \\ x_i \neq \epsilon}}^{\oplus} x_i \; - \sum_{\substack{i \in M \\ x_i \neq \epsilon}}^{\oplus'} x_i$$

is the range norm of x after only considering the finite components. Furthermore, if $x = \epsilon$, then $\delta(x) = \epsilon$ by definition.

It can been seen that if x is finite, then $\widetilde{\delta}(x) = \delta(x)$. Now using the above definition, we can modified Problem 2 as following:

Problem 3. Given a matrix $A \in \overline{\mathbb{R}}^{m \times n}$, solve

$$\begin{aligned} \widetilde{\delta}(b) \rightarrow \quad & \min \\ \text{subject to} \quad & b \in Im(A). \end{aligned}$$

The first problem corresponds to the case when only finite images of A are considered and the second problem corresponds to the case when the images of A are not necessarily finite.

Note that for the the second problem, we have not included the case when $b = \epsilon$ to be a solution to Problem 3; this is because $A \otimes \epsilon = \epsilon$ for any matrix A and $\delta(\epsilon) = \epsilon$ by definition, therefore it will always be a solution of Problem 3 and hence it will not make much sense to include $b = \epsilon$ to be a solution.

4.2.1 The Case when the Image Vector is Finite

We will start by investigating the case when only finite images of A are considered, i.e. Problem 2. In Chapter 2, we have discussed max-linear systems and we know that if the image vector is finite, it is sufficient to consider only the case when the matrix is doubly $\mathbb{R}$-astic. Before we can develop a solution method to solve Problem 2, we will need the following lemmas.

Lemma 4.2.1. [29] Suppose that $A \in \overline{\mathbb{R}}^{m\times n}$, $A \neq \epsilon$ and $b = A \otimes (A^* \otimes' 0)$ then

i) $b \leq 0$ and

ii) $b_i = 0$ for some $i \in M$.

Lemma 4.2.2. [29] Let $x, y \in \mathbb{R}^m$ and $\alpha \in \mathbb{R}$, then

i) $\delta(x \oplus y) \leq \delta(x) \oplus \delta(y)$ and

ii) $\delta(x) = \delta(\alpha \otimes x)$.

Using the above lemmas, we will obtain the following proposition which will present us a solution for Problem 2.

Proposition 4.2.1. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and $v_\alpha \in \mathbb{R}^m$ be a vector whose every component is equal to a constant $\alpha \in \mathbb{R}$, then $A \otimes (A^* \otimes' v_\alpha)$ is a solution to Problem 2.

Proof. Let $\bar{x} = A^* \otimes' 0$ and $\bar{b} = A \otimes \bar{x}$. Suppose that there exists a vector $y \in Im(A)$ such that $\delta(y) < \delta(\bar{b})$. By Lemma 4.2.1, the maximum value of $\bar{b}$ is equal to 0 and without loss of generality we can assume the maximum value of y is also equal to 0.
Let $\bar{b}_i$ be a minimum component of $\bar{b}$. Since $\delta(y) < \delta(\bar{b})$ and the maximum value of $\bar{b}$ and y are equal to 0, we have

$$\begin{aligned} 0 - \bar{b}_i > 0 - \min_{j=1,\dots,m} y_j \geq 0 - y_i, \\ y_i > \bar{b}_i. \end{aligned} \tag{4.1}$$

Since $y \in Im(A)$, therefore $\exists v \in \mathbb{R}^n$ such that $y = A \otimes v$. We will choose $k \in N$ such that

$$y_i = \max_{j=1,\dots,n} (a_{ij} + v_j) = a_{ik} + v_k. \tag{4.2}$$

Now, we can see that

$$\bar{b}_i = \max_{j=1,\dots,n} (a_{ij} + \bar{x}_j) \geq a_{ik} + \bar{x}_k. \tag{4.3}$$

Therefore, using (4.1), (4.2) and (4.3), we will get

$$a_{ik} + v_k > a_{ik} + \bar{x}_k.$$

It implies that $v_k > \bar{x}_k$. Note that

$$\bar{x}_k = \min_{i=1,\dots,m} (-a_{ik} + 0) = -a_{lk},$$

for some $l \in M$, then the inequality $v_k > \bar{x}_k$ will become

$$v_k > -a_{lk},$$
$$v_k + a_{lk} > 0.$$

But this implies that $y_l > 0$ and it is a contradiction to the assumption that the maximum value of y is equal to 0. Therefore $\bar{b}$ is a solution to Problem 2.

By Proposition 2.3.2, we can multiply the vector $\bar{b}$ by $\alpha \in \mathbb{R}$ such that $c \otimes \bar{b} \in Im(A)$. By Lemma 4.2.2, we know that $\delta(\bar{b}) = \delta(c \otimes \bar{b})$. Therefore by using the properties in (2.1), we will get

$$\begin{aligned}\hat{b} &= A \otimes (A^* \otimes' v_\alpha)\\ &= A \otimes (A^* \otimes' (0 \otimes \alpha))\\ &= (A \otimes (A^* \otimes' 0)) \otimes \alpha\\ &= \bar{b} \otimes \alpha\\ &= \alpha \otimes \bar{b}\end{aligned}$$

will also be a solution to Problem 2. $\square$

Example 4.2.1. Given a matrix $A = \begin{pmatrix} 6 & 2 & 3 \\ \epsilon & 5 & 6 \\ 7 & \epsilon & \epsilon \\ 10 & 11 & \epsilon \end{pmatrix}$. Find a vector $\bar{b}$ such that it is a solution to Problem 2.

Since A is doubly $\mathbb{R}$-astic, then by Proposition 4.2.1 a solution to Problem 2 is $\bar{b} = A \otimes$

$(A^* \otimes' 0)$.

$$\bar{x} = A^* \otimes' 0 = \begin{pmatrix} -6 & +\infty & -7 & -10 \\ -2 & -5 & +\infty & -11 \\ -3 & -6 & +\infty & +\infty \end{pmatrix} \otimes' 0$$
$$= \begin{pmatrix} -10 \\ -11 \\ -6 \end{pmatrix}$$

$$\bar{b} = A \otimes \bar{x} = \begin{pmatrix} 6 & 2 & 3 \\ \epsilon & 5 & 6 \\ 7 & \epsilon & \epsilon \\ 10 & 11 & \epsilon \end{pmatrix} \otimes \begin{pmatrix} -10 \\ -11 \\ -6 \end{pmatrix}$$
$$= \begin{pmatrix} -3 \\ 0 \\ -3 \\ 0 \end{pmatrix}$$

We can see that $\delta(\bar{b}) = 3$ and it immediately implies that $\delta(b) \geq 3, \forall b \in Im(A)$.

4.2.2 The Case when the Image Vector is Not Finite

Next we will investigate the case when the image vector may not be finite, therefore we will now starting to investigate Problem 3. As we have mentioned before, there are cases that eigenvectors of a matrix may not be finite. Using Theorem 3.6.2, we know that the set of finite eigenvectors is not empty if and only if the eigenvalues of all final classes are equal to the maximum cycle mean of the whole matrix. Therefore we will need to consider the case when some of the entries in the image vector is not finite.

We should also note that for some doubly $\mathbb{R}$-astic matrix, its image set will not only consists of finite vectors; there are image vectors which can contain ϵ entries. Let us start by considering the following example.

Example 4.2.2. Suppose $A = \begin{pmatrix} 4 & 6 & 7 \\ \epsilon & 2 & \epsilon \\ 5 & 2 & 6 \end{pmatrix}$. Find a vector $\bar{b}$ such that it is a solution to Problem 3.

Since A is doubly $\mathbb{R}$-astic, we can use Proposition 4.2.1 and we know that $A \otimes (A^* \otimes' 0)$ is a solution to Problem 2, i.e.

$$\begin{aligned} \bar{x} = A^* \otimes' 0 &= \begin{pmatrix} -4 & \infty & -5 \\ -6 & -2 & -2 \\ -7 & \infty & -6 \end{pmatrix} \otimes' 0 \\ &= \begin{pmatrix} -5 \\ -6 \\ -7 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} \bar{b} = A \otimes \bar{x} &= \begin{pmatrix} 4 & 6 & 7 \\ \epsilon & 2 & \epsilon \\ 5 & 2 & 6 \end{pmatrix} \otimes \begin{pmatrix} -5 \\ -6 \\ -7 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ -4 \\ 0 \end{pmatrix} \end{aligned}$$

Therefore $\widetilde{\delta}(\bar{b}) = \delta(\bar{b}) = 0 - (-4) = 4$. But if we let $x = (-5, \epsilon, -7)^T$; it means we can ignore the second column of A, then

$$\hat{b} = A \otimes x = \begin{pmatrix} 4 & 6 & 7 \\ \epsilon & 2 & \epsilon \\ 5 & 2 & 6 \end{pmatrix} \otimes \begin{pmatrix} -5 \\ \epsilon \\ -7 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ \epsilon \\ 0 \end{pmatrix}$$

and we will get $\widetilde{\delta}(\hat{b}) = 0$ and it is a solution to Problem 3.

The above example shows that the solution found by Proposition 4.2.1 which solve Problem 2 may not necessarily solve Problem 3.

Note that in order to obtain a solution of Problem 3 in the above example, we have set $x_2 = \epsilon$ and this is equivalent to removing the second column of matrix A and consider the image set of the new matrix. By doing this, the matrix will not be doubly $\mathbb{R}$-astic anymore. But using the argument we have discussed in section 2.3, we can remove the row in which it is equal to ϵ. Therefore this is equivalent to solving Problem 2 but with a reduced matrix.

In order to find a solution for Problem 3, we will need the following proposition.

Proposition 4.2.2. Let $A \in \overline{\mathbb{R}}^{m \times n}$, $A \neq \epsilon$, and $v_\alpha \in \mathbb{R}^m$ be a vector whose every component is equal to a constant α. If $x \in \overline{\mathbb{R}}^n$ such that $\bar{b} = A \otimes x$ is a solution to Problem 3 then $\forall j \in N$, either

i) $x_j = \epsilon$ or

ii) $x_j = (A^* \otimes' v_\alpha)_j$.

Proof. Suppose that A is not column $\mathbb{R}$-astic, i.e. $\exists j \in N$ such that $A_j = \epsilon$, it implies that we can choose any value for x_j as a solution to Problem 3 and therefore we can choose either

i) and ii) for x_j. Therefore, without loss of generality, we can assume A is column $\mathbb{R}$-astic.

Now, we will let $\bar{b}$ be a solution to Problem 3 and the set $E = \{i \in M \mid \bar{b}_i = \epsilon\}$.

Let us first consider the non-finite components of $\bar{b}$, we know that $\forall i \in E$,

$$\epsilon = \bar{b}_i = \max_{j=1,\dots,n} (a_{ij} + x_j).$$

It immediately follows that $\forall i \in E$, if $a_{ij} \neq \epsilon$ then $x_j = \epsilon$. We will let the set

$$F = \{j \in N \mid \exists i \in E \text{ such that } a_{ij} \neq \epsilon\}$$

and we can see that $x_j = \epsilon$, $\forall j \in F$. It remains to find the value of x_j when $j \notin F$.

Now let us consider the finite components of $\bar{b}$. Suppose that $i \notin E$, it implies that

$$\bar{b}_i = \max_{j=1,\dots,n} (a_{ij} + x_j) > \epsilon$$

and hence $\exists j \in N$ such that $a_{ij} \neq \epsilon$.

Also, A is column $\mathbb{R}$-astic, therefore $\forall j \notin F$, $\exists i \notin E$ such that $a_{ij} \neq \epsilon$. Now we shall consider the sub-matrix which is formed by deleting rows and columns of A with the indices E and F respectively. By the above two arguments, we can see that this sub-matrix is doubly $\mathbb{R}$-astic. Now we have transformed Problem 3 to Problem 2 but with a smaller matrix. Therefore, if we use Proposition 4.2.1 we know that $\forall j \notin F$,

$$x_j = \min_{i \notin E}(-a_{ij} + \alpha) = -\max_{i \notin E}(a_{ij} - \alpha).$$

Since $\forall j \notin F$, $a_{ij} = \epsilon, \forall i \in E$, it immediately follows that

$$\begin{aligned} x_j &= -\max_{i \notin E}(a_{ij} - \alpha), \\ &= -\max_{i \in M}(a_{ij} - \alpha), \\ &= \min_{i \in M}(-a_{ij} + \alpha), \\ &= (A^* \otimes' v_\alpha)_j. \end{aligned}$$

□

By the above proposition, we know that if x is a solution to Problem 3, there are only two values x can take for each component. Using this property, we can formulate an algorithm which finds a solution to Problem 3.

Suppose that $A \in \overline{\mathbb{R}}^{m \times n}$ and $x^{(k)} \in \overline{\mathbb{R}}^n$, $\forall k \in \mathbb{N}$. First we will assume that for all components in $x^{(0)}$, we have $x_j^{(0)} = (A^* \otimes' v_\alpha)_j$. Without loss of generality, we can assume that $\alpha = 0$, then we can let $x^{(0)} = A^* \otimes' 0$ and $b^{(0)} = A \otimes x^{(0)}$ be our initial solution.
Let $b_p^{(0)}$ be the minimum finite component of $b^{(0)}$. By Lemma 4.2.1, we know that

$$\max_{i=1,\ldots,m} b_i^{(0)} = 0$$

and therefore $\widetilde{\delta}(b^{(0)}) = -b_p^{(0)}$. If $b_p^{(0)} = 0$, it implies that $\widetilde{\delta}(b^{(0)}) = 0$ and it immediately follows that it is a solution to Problem 3. Let us suppose that $b_p^{(0)} \neq 0$, we know that

$$b_p^{(0)} = \max_{j=1,\ldots,n} (a_{pj} \otimes x_j) \text{ where } x_j = -\max_{i=1,\ldots,m} a_{ij}.$$

The next step will be to find which component of $x^{(0)}$ should be changed to ϵ in order to obtain a better solution. Note that when we let $x_j = \epsilon$ for some $j \in N$ in system (2.2), it is equivalent as disregarding/deleting the j^{th} column of A.

Another thing we need to be aware of is that $b^{(0)}$ may be a solution to Problem 3, but we do not know if this is the case at the moment. Therefore we will need to check for other possible solution and compare each of them.

Our aim now is to find some $j \in N$ such that if we will set $x_j^{(0)} = \epsilon$, the value of $b_p^{(0)}$ is increased so we may obtain a better range norm or $b_p^{(0)} = \epsilon$ so this value is not considered.

Let us look at the following matrix:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1r} & \dots & a_{1n} \\ . & . & . & . & . & . \\ a_{p1} & a_{p2} & \dots & a_{pr} & \dots & a_{pn} \\ . & . & . & . & . & . \\ a_{m1} & a_{m2} & \dots & a_{mr} & \dots & a_{mn} \end{pmatrix}.$$

If we delete a column from A, i.e. column r, then

$$\begin{aligned} b_p^{(1)} &= \max_{\substack{j=1,\dots,n \\ j \neq r}} (a_{pj} \otimes x_j) \\ &\leq b_p^{(0)}. \end{aligned}$$

From the inequality above, we can see that we cannot increase the value of $b_k^{(0)}$ by deleting columns of A. Therefore we need to find a new vector, namely $x^{(1)}$, such that $b_p^{(1)} = (A \otimes x^{(1)})_p$ is equal to ϵ.

In order for $b_p^{(1)}$ to be equal to ϵ, i.e.

$$a_{p1} \otimes x_1^{(1)} \oplus a_{p2} \otimes x_2^{(1)} \oplus \dots \oplus a_{pn} \otimes x_n^{(1)} = \epsilon,$$

we will let

$$x_j^{(1)} = \begin{cases} \epsilon & \text{if } a_{pj} \in \mathbb{R}, \\ x_j^{(0)} & \text{otherwise.} \end{cases}$$

If $a_{pj} \in \mathbb{R}, \forall j \in N$, then we will need to set $x^{(1)} = \epsilon$ so $b_p^{(1)} = \epsilon$. But then, we know that $b^{(1)} = \epsilon$ and we have disregarded this to be a solution for Problem 3 in the formulation of this problem. Therefore, we can conclude that we cannot obtain a better solution than $b^{(0)}$ and hence, $b^{(0)}$ is a solution to Problem 3.

Note that $b^{(1)}$ may or may not be a better solution than $b^{(0)}$; even in the case when $b^{(1)}$ is a better solution than $b^{(0)}$, we cannot tell if it is a solution to Problem 3. We have to repeat the above process and obtain $b^{(2)}$, $b^{(3)}$, etc until we either found a $l \in \mathbb{N}$ such that $\widetilde{\delta}(b^{(l)}) = 0$ or $x^{(l)} = \epsilon$. In both cases, we will stop and in the first case, it immediately follows that $b^{(l)}$ is a solution to Problem 3. In the second case, we have obtained all possible candidates for a solution to Problem 3, then if we find $\bar{b}$ such that

$$\widetilde{\delta}(\bar{b}) = \min_{i=1,\dots,l-1}(\widetilde{\delta}(b^{(i)})),$$

$\bar{b}$ will be a solution to our problem.

We should also note that $l \leq n$, this is because the number of components equal to ϵ in x is increased by at least one after each iteration. Therefore the vector x must be equal to ϵ after n iterations and hence we will only need to find at most n vectors.

Now we can generate the following algorithm to perform the steps we discussed above.

Algorithm 1.

Input: $A \in \overline{\mathbb{R}}^{m \times n}$

Output: $\bar{b} \in \overline{\mathbb{R}}^{m}$, a solution to Problem 3

Set $x^{(0)} := A^* \otimes' 0$, $b^{(0)} := A \otimes x^{(0)}$ and $\bar{b} := b^{(0)}$.

For $k = 1$ to $n - 1$ do

Begin

If $\widetilde{\delta}(b^{(k)}) = 0$, then $\bar{b}$ is optimal. Stop.

Find p such that

$$b_p^{(k)} = \min_{i=1,\ldots,m} b_i^{(k)} > \epsilon.$$

For $j = 1$ to n do

If $a_{pj} \in \mathbb{R}$, set $x_j^{(k+1)} := \epsilon$, else set $x_j^{(k+1)} := x_j^{(k)}$.

If $x^{(k+1)} = \epsilon$, $\bar{b}$ is optimal. Stop.

Set $b^{(k+1)} := A \otimes x^{(k+1)}$.

If $\widetilde{\delta}(b^{(k+1)}) < \widetilde{\delta}(b^{(k)})$, set $\bar{b} := b^{(k+1)}$.

End

4.3 Maximizing the Range Norm

In the previous section, we have investigated the problem of minimizing the range norm of an image vector; now we will move on to the case when we want to maximize the range norm. In some real-life systems, the manufacturers will not prefer to start all their machines simultaneously but rather to be as spread out as possible.

One of the reason of this being the case is because the machines in manufacturing are likely to require a huge amount of powers in order for them to work efficiently; if all the machines starts simultaneously, this will create a huge power surge and the power circuit may not be able to handle it, this will cause the circuit to break and in the worst case scenario, a fire or even an explosion may occur.

This provides the motivation to investigate the case when we are maximizing the range norm of an image set. This can be formulated as the following problem.

Problem 4. Given a matrix $A \in \overline{\mathbb{R}}^{m \times n}$, solve

$$\begin{aligned} \delta(b) \rightarrow \quad & \max \\ \text{subject to} \quad & b \in Im(A). \end{aligned}$$

4.3.1 The Case when the Matrix is Finite

We will use a similar approach as in the previous section, first we will consider the case when the matrix A is finite.

Proposition 4.3.1. Let $A \in \mathbb{R}^{m \times n}$, then $\forall b \in Im(A)$

$$\begin{aligned} \delta(b) &\leq \max_{j=1,\ldots,n} \delta(A_j) \\ &= \max_{j=1,\ldots,n} \big(\max_{i=1,\ldots,m} a_{ij} - \min_{i=1,\ldots,m} a_{ij} \big) \end{aligned} \tag{4.4}$$

Proof. Let $b \in Im(A)$, b_r be the maximum value of b and b_l be the minimum value of b. Suppose that

$$\delta(b) > \max_{j=1,\ldots,n} \delta(A_j),$$

then it implies that $\forall j \in N$,

$$\begin{aligned} b_r - b_l &> \max_{i=1,\ldots,m} a_{ij} - \min_{i=1,\ldots,m} a_{ij} \\ &\geq a_{rj} - a_{lj}. \end{aligned}$$

Hence we have $\forall j \in N$,

$$a_{lj} - b_l > a_{rj} - b_r \tag{4.5}$$

and this implies $r \notin M_j(A, b)$, $\forall j \in N$. But $b \in Im(A)$ and by Corollary 2.3.1 and

Proposition 2.3.1, we know that

$$\bigcup_{j \in N} M_j(A, b) = M$$

and this is a contradiction. Therefore

$$\delta(b) \leq \max_{j=1,\ldots,n} \delta(A_j).$$

$\square$

Since the range norm of the image vector is bounded by (4.4), therefore if there exists $b \in Im(A)$ such that

$$\delta(b) = \max_{j=1,\ldots,n} \delta(A_j),$$

b will be a solution to Problem 4. Note that $\forall j \in N$, $\exists x \in \overline{\mathbb{R}}$ such that $A_j = A \otimes x$ and therefore $A_j \in Im(A)$, $\forall j \in N$. Hence for any finite matrices, we can find a $k \in N$ such that

$$\delta(A_k) = \max_{j=1,\ldots,n} \delta(A_j)$$

and A_k will be a solution to Problem 4.

Example 4.3.1. Given $A = \begin{pmatrix} 4 & 3 & 8 \\ 3 & 2 & -1 \\ 5 & 2 & 6 \\ -1 & 3 & 0 \end{pmatrix}$. Find a vector $\bar{b}$ such that it is a solution to Problem 4.

Since A is a finite matrix, by Proposition 4.3.1 we know that

$$\delta(\bar{b}) = \delta(A_k) = \max_{j=1,\ldots,n} \delta(A_j)$$

and therefore

$$\max_{j=1,2,3} \delta(A_j) = \max(5-(-1), 3-2, 8-(-1)) = 9 = \delta(A_3).$$

Let $x = (\epsilon, \epsilon, 0)^T$, then $A \otimes x = A_3$ will be a solution to Problem 4.

4.3.2 The Case when the Matrix is Non-Finite

Finally we will consider the case when we are maximizing the range norm from the image set of a non-finite matrix.

Proposition 4.3.2. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and non-finite, i.e. $\exists i \in M$, $j \in N$ such that $a_{ij} = \epsilon$, then Problem 4 is unbounded.

Proof. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and $\exists r \in M$, $s \in N$ such that $a_{rs} = \epsilon$. Let $x \in \mathbb{R}^n$ such that

$$x_j = \begin{cases} \alpha & \text{if } j = s, \\ 0 & \text{otherwise.} \end{cases}$$

where $\alpha \in \mathbb{R}$ is an arbitrary constant. Let $b = A \otimes x$, then it immediately follows that $\forall i \in M$,

$$b_i = (A \otimes x)_i = \max(\max_{j \neq s}(a_{ij}), \alpha + a_{is}).$$

Since $a_{rs} = \epsilon$ and $\alpha + a_{rs} = \epsilon$, $\forall \alpha \in \mathbb{R}$, we then have

$$b_r = \max(\max_{j\neq s}(a_{rj}), \alpha + a_{rs}) = \max_{j \neq s}(a_{rj}).$$

Since A is row $\mathbb{R}$-astic, then $\exists k \in N$ such that $a_{rk} \in \mathbb{R}$ and we have

$$b_r = \max_{j\neq s}(a_{rj}) = a_{rk}.$$

Similarly A is column $\mathbb{R}$-astic, then $\exists l \in M$, $l \neq r$ such that $a_{ls} \in \mathbb{R}$. If we choose a sufficiently large α, we then have

$$b_l = \max(\max_{j \neq s}(a_{lj}), \alpha + a_{ls}) = \alpha + a_{ls}.$$

Since

$$\delta(b) \geq b_l - b_r = \alpha + a_{ls} - a_{rk}$$

and α can be arbitrarily large, we have $b_l - b_r$ is unbounded and hence $\delta(b)$ is unbounded. $\square$

4.4 Summary

In this chapter, we have investigated on minimization and maximization range norm of vectors from the image set of a matrix. We have developed an exact solution method on solving the case of minimizing range norm when the image of a matrix is finite. We have also provided an algorithm for the case when we do not restrict the image to be a finite vector. We have also shown that for finite matrices, we can solve the maximization case by considering each column of the matrix and finding the column which has the maximum range norm. And for the case of non-finite matrices, we have proved that the maximization case is unbounded.

Chapter 5

Optimizing Range Norm of the Image Set With Prescribed Components

5.1 Introduction

In the previous chapter, we have investigated the problem of minimization and maximization of the range norm over an image set. In this chapter we will continue to investigate this problem but with additional constraint.

In some cases, the manufacturers will have a preference on the starting times for some of their machines. Therefore the vector of starting times for the corresponding components is prescribed. We will suppose that these prescribed starting times are feasible, i.e. they are components of some eigenvectors, then the manufacturers will want to find the starting times for the remaining machines so the vector of starting times for all the machines will be an eigenvector.

On top of that the manufacturers may have some other criteria on the structures of the starting times. One of these criteria could be to choose a starting times for the other machines such that the range norm of all starting times is minimized or maximized. This is due to the reasons we have discussed in the previous chapter. We will first investigate the problem of minimization, then we will show that the method we developed for the minimization can be modified to solve the maximization case.

5.2 Minimizing the Range Norm

We shall first consider the case of minimization. Let us consider a matrix $A \in \overline{\mathbb{R}}^{m\times n}$, if we permute the row and column indexes simultaneously, this is equivalent to renumbering the machines in the system and therefore the overall structure is not affected. If we suppose that the number of prescribed components to be p, then we can assume without loss of generality, that the machines with predetermined starting times are Machine 1 to Machine p and we will let $P = \{1, ..., p\}$. For simplicity we will separate the matrix A into two matrices and they are defined as follow.

Definition 5.2.1. Let $A \in \overline{\mathbb{R}}^{m\times n}$, and $1 \leq p \leq m - 1$. We will denote A^p to be the matrix that consists of the first p row(s) of A and $\bar{A}^p$ to be the matrix generated by deleting the first p row(s) of A.

We could see that the matrix A^p consists of the upper part of A whereas $\bar{A}^p$ consists of the lower part of A. We will assume without loss of generality, the following:

- The vector of prescribed components is finite and it is an image of the upper matrix, namely A^p, i.e. $c \in Im(A^p) \cap \mathbb{R}^p$.
- The undetermined starting time is also finite.

Then the above problem can be formulated as below:

Problem 5. Given $A \in \overline{\mathbb{R}}^{m\times n}$ and $c \in Im(A^p) \cap \mathbb{R}^p$, $1 \leq p \leq m - 1$, find $d \in \mathbb{R}^{m-p}$ such that

$$\begin{aligned} \delta(b) \rightarrow \quad & \min \\ \text{subject to} \quad & b \in Im(A) \\ & b^T = (c^T, d^T). \end{aligned}$$

Note that if $p = 0$, then the problem will be the same as Problem 2 in Chapter 4. Similarly if the starting times for all the machines are prescribed, i.e. $p = m$, then there is nothing to be determined, therefore we can assume without loss of generality, that $1 \leq p \leq m - 1$.

We shall also note that both c and d are finite and therefore b is finite, henceforth only finite images of A are considered and by the same argument as in the previous chapters, we only need to consider the case when A is doubly $\mathbb{R}$-astic.

We will for simplicity, in the rest of this chapter, call $d \in \mathbb{R}^{m-p}$ to be a *feasible solution* to Problem 5 if the vector $b \in Im(A)$ where $b^T = (c^T, d^T)$. We will also call $d \in \mathbb{R}^{m-p}$ to be an *optimal solution* if d solves Problem 5.

5.2.1 The Case when Only One Machine is Prescribed

Before we investigate the problem for all cases of p, we would like to consider the two special cases for this problem; when $p = 1$ and $p = m - 1$. Let us consider the following proposition.

Proposition 5.2.1. Let $A \in \overline{\mathbb{R}}^{m \times n}$ be doubly $\mathbb{R}$-astic and $c \in Im(A^p) \cap \mathbb{R}^p$, $1 \leq p \leq m - 1$. Suppose that $\bar{b} \in \mathbb{R}^m$ is an optimal solution to Problem 2 for A and $d \in \mathbb{R}^{m-p}$ is an optimal solution to Problem 5 for A and c, then $\delta(\bar{b}) \leq \delta(b)$ where $b^T = (c^T, d^T)$.

Proof. Since $b \in Im(A)$ and $\bar{b}$ has the minimum range norm over the images of A, then it immediately follows that $\delta(\bar{b}) \leq \delta(b)$. $\square$

Proposition 5.2.1 provides a lower bound for the range norm of the vector obtained from an optimal solution of Problem 5. Using this result, we can immediately solve the case when

$p = 1$. Suppose that only the starting time of one machine is prescribed, then by Proposition 4.2.1 and Proposition 5.2.1, we will obtain the following result.

Proposition 5.2.2. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and $c = (c_1) \in \mathbb{R}$ be the vector of prescribed component. Suppose that $b = A \otimes (A^* \otimes' 0)$ and $d = (c_1 \otimes b_1^{-1}) \otimes b$ where b_1 is the first component of b. Then the vector $(d_2, ..., d_m)^T$ is an optimal solution to Problem 5.

Proof. Since $d_1 = (c_1 \otimes b_1^{-1}) \otimes b_1 = c_1$, then $(d_2, ..., d_m)^T$ is a feasible solution for Problem 5. We also know that b is an optimal solution to Problem 2 by Proposition 4.2.1, therefore we can see that any multiple of b is also an optimal solution to Problem 2, i.e. $\delta(b) = \delta(d)$. By Proposition 5.2.1 we have obtained the lower bound for Problem 5 and since d is a feasible solution to this problem, it immediately follows that $(d_2, ..., d_m)^T$ is an optimal solution to Problem 5. □

The above proposition provides us an optimal solution to Problem 5 when $p = 1$ and it is not difficult to see that for the case when $p = 1$, we can always obtain an optimal solution to Problem 5 by finding an optimal solution to Problem 2.

5.2.2 The Case when All but One Machine are Prescribed

Now we will consider the next special case; when $p = m - 1$. This is the case when all but one of the starting time are prescribed and the method is less straight forward than the previous case.

First we will need to obtain a feasible solution for d. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be a given doubly $\mathbb{R}$-astic matrix and $c \in Im(A^p) \cap \mathbb{R}^p$ be the vector of prescribed components, then $\exists \bar{x} \in \overline{\mathbb{R}}^n$ where $\bar{x} = (A^{m-1})^* \otimes' c$ such that

$$c = A^{m-1} \otimes \bar{x}.$$

Using the vector $\bar{x}$, we can obtain

$$\bar{d} = \bar{A}^{m-1} \otimes \bar{x}.$$

We can see that $\bar{d}$ is a feasible solution for Problem 5 from the following equation.

$$A \otimes \bar{x} = \begin{pmatrix} A^{m-1} \\ \bar{A}^{m-1} \end{pmatrix} \otimes \bar{x} = \begin{pmatrix} A^{m-1} \otimes \bar{x} \\ \bar{A}^{m-1} \otimes \bar{x} \end{pmatrix} = \begin{pmatrix} c \\ \bar{d} \end{pmatrix} = b.$$

Suppose that L and U represent the minimum and the maximum value of c respectively, i.e.

$$L = \min_{i=1,\dots,m-1} c_i,$$
$$U = \max_{i=1,\dots,m-1} c_i.$$

We know that $\bar{d}$ consists of only one element, therefore there are only three cases to consider:

$$\begin{array}{lll} 1) & L \le \bar{d} \le U, & \text{i.e.} \quad \delta(b) = \delta(c) = U - L \\ 2) & \bar{d} < L, & \text{i.e.} \quad \delta(b) = U - \bar{d} \\ 3) & U < \bar{d}, & \text{i.e.} \quad \delta(b) = \bar{d} - L \end{array}$$

In the first case, we see that we cannot improve the range norm by replacing the value of $\bar{d}$ with any larger or smaller value, therefore $\bar{d}$ is an optimal solution of Problem 5. Next we will consider the second case.

Proposition 5.2.3. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic, $c \in Im(A^p) \cap \mathbb{R}^p$ and $\widetilde{d}$ be an optimal solution to Problem 5. Suppose that $\bar{x} = (A^p)^* \otimes' c$ and $\bar{d} = \bar{A}^p \otimes \bar{x}$, then $\widetilde{d} \le \bar{d}$.

Proof. Let $x \in \overline{\mathbb{R}}^n$ be any solution to the linear system $A^p \otimes x = c$. Since $\bar{x} = (A^p)^* \otimes' c$ is the principal solution to the system $A^p \otimes x = c$, therefore by Theorem 2.3.1, we know that

$$x \leq \bar{x}. \tag{5.1}$$

Let $\widetilde{d}$ be an optimal solution to Problem 5, then $\exists \widetilde{x} \in \overline{\mathbb{R}}^n$ such that $\widetilde{d} = \bar{A}^p \otimes \widetilde{x}$ and $A^p \otimes \widetilde{x} = c$. But by (5.1), we know that $\widetilde{x} \leq \bar{x}$ and therefore

$$\widetilde{d} = \bar{A}^p \otimes \widetilde{x} \leq \bar{A}^p \otimes \bar{x} = \bar{d}.$$

□

From the above proposition we have found the upper bound for d and it is $\bar{d}$. If we look back at the second case, we can see that we cannot improve the range norm by replacing $\bar{d}$ with any smaller value, therefore we can deduce that $\bar{d}$ is an optimal solution to Problem 5 for this case.

Finally we will consider the case when $U < \bar{d}$. For this situation, our goal is to find out a smaller value to replace $\bar{d}$, ideally so that this value is between L and U. This can be quickly checked by letting $d^{(1)} = U$ and find out if the vector $\begin{pmatrix} c \\ U \end{pmatrix}$ is an image of A. If this is the case, then $d^{(1)} = U$ is an optimal solution to Problem 5.

Suppose that this is not the case, i.e. $U < d^{(k)}$, $k = 1, 2, ...$, then we would want to find out what is the smallest value we can replace $\bar{d}$ without affecting feasibility and this value will be then our optimal solution to Problem 5. Let us consider the value $\bar{d}$, we know that $\bar{d}$ is found by the following:

$$\begin{aligned} \bar{d} &= \bar{A}^{m-1} \otimes \bar{x} \\ &= (a_{m1} \otimes \bar{x}_1) \oplus (a_{m2} \otimes \bar{x}_2) \oplus ... \oplus (a_{mn} \otimes \bar{x}_n) \\ &= \max_{j=1,\dots,n} (a_{mj} + \bar{x}_j). \end{aligned}$$

In order to find another feasible solution, namely $d^{(1)}$ such that $d^{(1)} < \bar{d}$, we will need to decrease some or all of the values in $\bar{x}$. Suppose that

$$K = \{k \in N \mid a_{mk} + \bar{x}_k = \max_{j=1,\ldots,n}(a_{mj} + \bar{x}_j) = \bar{d}\},$$

then we will choose a new x which will be called $x^{(1)}$ as the following,

$$x_j^{(1)} = \begin{cases} \bar{x}_j - \tau^{(1)} & \text{if } j \in K, \\ \bar{x}_j & \text{otherwise} \end{cases}$$

where $\tau^{(1)} > 0$ and we will let

$$d^{(1)} = \max_{j=1,\ldots,n}(a_{mj} + x_j^{(1)}).$$

Now we will need to check whether $d^{(1)}$ is a feasible solution to the problem. Let us consider the following linear system, $A^p \otimes x = b$, we will let

$$P_j = \{q \in P \mid (a_{qj} - c_q) = \max_{i=1,\ldots,p}(a_{ij} - c_i)\} \ \ \forall j \in N.$$

From Theorem 2.3.1 we know that x is a solution to the system if and only if

i) $x \leq \bar{x}$ and

ii) $\bigcup_{j \in N_z} P_j = P$ where $N_z = \{j \in N \mid x_j = \bar{x}_j\}$.

Using this we can find out whether $x^{(1)}$ is a solution to the system by checking if i) and ii) are satisfied. We know that i) is automatically satisfied by the choice of $x^{(1)}$. We also know that $x_j^{(1)} = \bar{x}_j$ if $j \notin K$, therefore we can check if ii) is satisfied by finding whether

$$\bigcup_{j \notin K} P_j = P$$

is true.

If ii) is not satisfied, it means that $d^{(1)}$ is not a feasible solution and we cannot reduce the value of $\bar{d}$. On the other hand, if ii) is satisfied, it means that $\forall j \in K$, the values for $x_j^{(1)}$ can be decreased arbitrarily without affecting feasibility, i.e. we can take $\tau^{(1)}$ to be any arbitrary positive value. We will want to take the smallest possible value for $d^{(1)}$, therefore we will need to choose a value for $\tau^{(1)}$ such that

$$\max_{j=1,\ldots,n} (a_{mj} + x_j^{(1)})$$

is minimized. Hence we will need $\forall k \in K$,

$$\begin{aligned}\max_{\substack{j=1,\ldots,n \\ j \notin K}} (a_{mj} + x_j^{(1)}) &\geq a_{mk} + x_k^{(1)}, \\ \max_{\substack{j=1,\ldots,n \\ j \notin K}} (a_{mj} + \bar{x}_j) &\geq a_{mk} + \bar{x}_k - \tau^{(1)},\end{aligned}$$

i.e.

$$\tau^{(1)} \geq (a_{mk} + \bar{x}_k) - \max_{\substack{j=1,\ldots,n \\ j \notin K}} (a_{mj} + \bar{x}_j) > 0. \tag{5.2}$$

Since the value of $d^{(1)}$ is minimized when we take any value for $\tau^{(1)}$ satisfying the inequality (5.2), therefore for simplicity we will choose the smallest possible value for $\tau^{(1)}$, i.e. we will let

$$\tau^{(1)} = (a_{mk} + \bar{x}_k) - \max_{\substack{j=1,\ldots,n \\ j \notin K}} (a_{mj} + \bar{x}_j). \tag{5.3}$$

Note that if we let $W = \{a_{mj} + \bar{x}_j \mid j = 1, ..., n\}$, then $\tau^{(1)}$ is the difference between the largest value and the second largest value in W. It may be possible that the second largest value in W is equal to ϵ, i.e.

$$\max_{\substack{j=1,...,n \\ j \notin K}} (a_{mj} + \bar{x}_j) = \epsilon.$$

This implies that we can set $d^{(1)}$ to be arbitrarily small without affecting feasibility and hence we can set $d^{(1)} = U$ to be our optimal solution. But this contradicts our assumption that $d^{(1)} = U$ is not a feasible solution earlier on, therefore we know that this cannot be the case here.

By finding $\tau^{(1)}$, we have obtained the minimum value for $d^{(1)}$. The next step will be to find out if we can replace $d^{(1)}$ with an even smaller value and therefore we will repeat the process again and use $x^{(1)}$ to find $x^{(2)}$, $\tau^{(2)}$ and $d^{(2)}$, etc. We will stop when we cannot obtain a better feasible solution for d and this will be an optimal solution for Problem 5.

There are several points we should be aware for this method. The first is when we are checking the feasibility of $d^{(r)}$, $\forall r \in \mathbb{N}$, we can see that i) is automatically satisfied again because of the choice of $x^{(r)}$. Therefore in each step, we only need to check that if ii) is satisfied.

Secondly, note that the feasibility can also be checked by calculating $A^p \otimes x^{(r)}$ and find out if it is equals to c. This will require $O(pn^2)$ operations. But when we are using our method of checking feasibility, the same sets $P_1, ..., P_n$ are used repeatedly so they are only needed to be calculated once. Also finding the union of these sets and checking if it is equal to P only requires $O(pn)$ operations. Therefore this method is more efficient than calculating $A^p \otimes x^{(r)}$ every time we obtained a new $x^{(r)}$.

Finally we know that due to the choice of $\tau^{(1)}$ (5.3), the number of elements in the set K

will increase by at least one from the previous step. We also know that if $K = N$ then

$$\bigcup_{j \notin K} P_j = \emptyset \neq P.$$

This means that $d^{(n)}$ is infeasible. Therefore the maximum number of steps required will be less than or equal to n.

After considering the case when $p = m - 1$, we have developed a method for solving this case. In fact, it is possible to modify this method to solve the general case, which we will investigate below.

5.2.3 The General Case

Now we will start to consider the general case for p. We will modify the method we developed for the case when $p = m - 1$ so that it will generate a vector for the case when $2 \leq p \leq m - 1$. And we will show that the vector generated by this method is an optimal solution to Problem 5.

Suppose that $A \in \overline{\mathbb{R}}^{m\times n}$ be a given matrix and $c \in Im(A^p) \cap \mathbb{R}^p$, $1 \leq p \leq m-1$, be the vector of prescribed components. We will do the same as before, we will find $\bar{x} = (A^p)^* \otimes' c$ and calculate $\bar{d} = \bar{A}^p \otimes \bar{x}$. We will also find the sets $P_1, ..., P_n$ such that $\forall j \in N$,

$$P_j = \{q \in P \mid (a_{qj} - q_k) = \max_{i=1,\dots,p} (a_{ij} - c_i)\}.$$

Similarly as before, we will let U be the maximum value of c, i.e.

$$U = \max_{i=1,\dots,p} c_i,$$

and $\bar{d}_{max}$ represents the maximum value of $\bar{d}$, i.e.

$$\bar{d}_{max} = \max_{i=1,...,m-p} \bar{d}_i.$$

If $\bar{d}_{max} > U$ then the range norm may be improved by replacing the corresponding entries in $\bar{d}$ with a smaller value. Therefore we will let

$$R = \{r \in \{1, ..., m-p\} \mid \bar{d}_r = \bar{d}_{max}\}$$

and in addition, we will let

$$K = \{k \in N \mid \exists r \in R, \bar{d}_r = \max_{j=1,...,n} (a_{r+p,j} + \bar{x}_j) = a_{r+p,k} + \bar{x}_k\}.$$

We will also set a new x which will be called $x^{(1)}$ such that

$$x_j^{(1)} = \begin{cases} \bar{x}_j - \tau^{(1)} & \text{if } j \in K \\ \bar{x}_j & \text{otherwise} \end{cases}$$

where $\tau^{(1)} > 0$.

The next step will be checking the feasibility of the new value. We will check the feasibility by the same method as before. If

$$\bigcup_{j \notin K} P_j = P$$

is false, then the new value is not feasible and $\bar{d}$ will be our optimal solution to Problem 5. Otherwise it is feasible and we will need to determine a value for $\tau^{(1)}$.

Our objective here is to find a minimum value for $\tau^{(1)}$ such that we cannot improve the range norm of the resulting vector by taking any value of $\tau^{(1)}$ larger than this lower bound. Since we need to consider all entries in the new d, therefore this process will be less straightforward than the case when $p = m - 1$.

First let us consider how we obtain the new d. We will let $d^{(1)} = \bar{A}^p \otimes x^{(1)}$, then $\forall i \in \{1, ..., m-p\}$

$$\begin{aligned} d_i^{(1)} &= \max_{j=1,\ldots,n} (a_{i+p,j} + x_j^{(1)}) \\ &= \max \big(\max_{\substack{j=1,\ldots,n \\ j \notin K}} (a_{i+p,j} + x_j^{(1)}), \max_{k \in K} (a_{i+p,k} + x_k^{(1)}) \big) \\ &= \max \big(\max_{\substack{j=1,\ldots,n \\ j \notin K}} (a_{i+p,j} + \bar{x}_j), \max_{k \in K} (a_{i+p,k} + \bar{x}_k - \tau^{(1)}) \big). \end{aligned} \tag{5.4}$$

We also know that $\forall r \in R$,

$$\begin{aligned} d_r^{(1)} &\geq \max_{k \in K} (a_{r+p,k} + \bar{x}_k - \tau^{(1)}) \\ &\geq \bar{d}_{max} - \tau^{(1)}. \end{aligned} \tag{5.5}$$

Let $d_s^{(1)}$ be one of the maximum values of $d^{(1)}$, from (5.4) we can see that this value is minimized when $\tau^{(1)}$ is large enough so that

$$\max_{\substack{j=1,\ldots,n \\ j \notin K}} (a_{s+p,j} + \bar{x}_j)) \geq \max_{k \in K} (a_{s+p,k} + \bar{x}_k - \tau^{(1)}).$$

Therefore

$$d_s^{(1)} = \max_{\substack{j=1,\ldots,n \\ j \notin K}} (a_{s+p,j} + \bar{x}_j). \tag{5.6}$$

Now using (5.5) and (5.6), we know that for sufficiently large $\tau^{(1)}$, we have

$$\begin{aligned}\bar{d}_{max} - \tau^{(1)} &\leq d_r^{(1)}, \quad \forall r \in R,\\ &\leq \max_{i=1,\dots,m-p} d_i^{(1)},\\ &= \max_{i=1,\dots,m-p} \Big(\max_{\substack{j=1,\dots,n\\ j\notin K}} (a_{i+p,j} + \bar{x}_j)\Big).\end{aligned}$$

Hence

$$\tau^{(1)} \geq \bar{d}_{max} - \max_{i=p+1,\dots,m} \Big(\max_{\substack{j=1,\dots,n\\ j\notin K}} (a_{ij} + \bar{x}_j)\Big) > 0.$$

We will again choose the smallest possible value as our $\tau^{(1)}$, i.e. we will let

$$\tau^{(1)} = \bar{d}_{max} - \max_{i=p+1,\dots,m} \Big(\max_{\substack{j=1,\dots,n\\ j\notin K}} (a_{ij} + \bar{x}_j)\Big). \tag{5.7}$$

Now if we look at (5.7) and let $p = m - 1$, then we can see that (5.7) coincides with (5.3). We can also see that it may be possible that

$$\max_{i=p+1,\dots,m} \Big(\max_{\substack{j=1,\dots,n\\ j\notin K}} (a_{ij} + \bar{x}_j)\Big) = \epsilon$$

and it immediately implies that we can set the maximum value of $d^{(1)}$ arbitrarily small. Therefore we can set $\tau^{(1)} = \bar{d}_{max} - U$ and the resulting $d^{(1)}$ will be an optimal solution to Problem 5.

It may also be the case that the maximum value of $d^{(1)}$ is smaller or equal to U. If this is the case, we will let

$$\eta = U - \max_{i=1,\dots,m-p} d_i^{(1)}$$

and the vector $\eta \otimes d^{(1)}$ will be our optimal solution to Problem 5. Otherwise we will repeat the process again and replace $\bar{x}$ and $\bar{d}$ with $x^{(1)}$ and $d^{(1)}$ respectively and find $x^{(2)}$, $d^{(2)}$ and

$\tau^{(2)}$, etc.

Algorithm 2.
Input: $A \in \overline{\mathbb{R}}^{m\times n}$ and $c \in \mathbb{R}^p$, $2 \leq p \leq m-1$
Output: $\hat{d} \in \mathbb{R}^{m-p}$, an optimal solution to Problem 5.

Set $x^{(0)} := (A^p)^* \otimes' c$, $d^{(0)} := \bar{A}^p \otimes x^{(0)}$,

$$d^{(0)}_{max} := \max_{i=1,\dots,m-p} d^{(0)}_i$$

and

$$U := \max_{i=1,\dots,p} c_i.$$

If $d^{(0)}_{max} \leq U$
$d^{(0)}$ cannot be improved and therefore $\hat{d} := d^{(0)}$. Stop.
For all $j \in N$, find

$$P_j = \{q \in P \mid (a_{qj} - q_k) = \max_{i=1,\dots,p} (a_{ij} - c_i)\}.$$

Set $u := 0$.
While $d^{(u)}_{max} > U$ do
Begin
Find

$$R = \{r \in \{1, ..., m-p\} \mid d^{(u)}_r = d^{(u)}_{max}\}$$

and

$$K = \{k \in N \mid \exists r \in R, \bar{d}_r = \max_{j=1,\dots,n} (a_{r+p,j} + \bar{x}_j) = a_{r+p,k} + \bar{x}_k\}.$$

If

$$\bigcup_{j \notin K} P_j = P$$

Set $x^{(u+1)}$ by the following:

$$x^{(u+1)}_j := \begin{cases} x^{(u)}_j - \tau^{(u+1)} & \text{if } j \in K, \\ x^{(u)}_j & \text{otherwise} \end{cases}$$

where

$$\tau^{(u+1)} := d^{(u)}_{max} - \max_{i=p+1,\dots,m} \Big(\max_{\substack{j=1,\dots,n \\ j \notin K}} (a_{ij} + x^{(u)}_j) \Big).$$

If $\tau^{(u+1)} = +\infty$, then we will set $\tau^{(u+1)} := d_{max}^{(u)} - U$.
Set $d^{(u+1)} := \bar{A}^p \otimes x^{(u+1)}$ and

$$d_{max}^{(u+1)} := \max_{i=1,\dots,m-p} d_i^{(u+1)}.$$

If $d_{max}^{(u+1)} \leq U$
Set $\eta := U - d_{max}^{(u+1)}$ and $d^{(u+1)} := \eta \otimes d^{(u+1)}$.
Then $\hat{d} := d^{(u+1)}$ will be our optimal solution. Stop.
Else
Set $u := u + 1$.
Else
$d^{(u)}$ cannot be improved anymore and therefore $\hat{d} := d^{(u)}$. Stop.
End

5.2.4 Correctness of the Algorithm

Now we need to show that Algorithm 2 produce an optimal solution to Problem 5. First let us consider the following proposition.

Proposition 5.2.4. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and $c \in \mathbb{R}^p$ be the vector of prescribed components. Suppose that $d^{(1)}, d^{(2)}, ..., d^{(l)}$ be all the d vectors produced by Algorithm 2, then

$$\delta \begin{pmatrix} c \\ d^{(1)} \end{pmatrix} \geq \delta \begin{pmatrix} c \\ d^{(2)} \end{pmatrix} \geq ... \geq \delta \begin{pmatrix} c \\ d^{(l)} \end{pmatrix}.$$

Proof. Suppose that L and U represent the minimum and the maximum value of b respectively, i.e.

$$L = \min_{i=1,\dots,p} c_i,$$

$$U = \max_{i=1,\dots,p} c_i.$$

and $\forall u \in \{1, ..., l\}$, $d_{min}^{(u)}$ and $d_{max}^{(u)}$ represent the minimum and the maximum value of $d^{(u)}$

respectively, i.e.

$$d_{min}^{(u)} = \min_{i=1,\ldots,m-p} d_i^{(u)},$$
$$d_{max}^{(u)} = \max_{i=1,\ldots,m-p} d_i^{(u)}.$$

From Algorithm 2, we know that $\forall u$, $d_{max}^{(u)} \geq U$. Therefore if we consider the u^{th} and the $u+1^{th}$ term, then we will have

$$\begin{aligned} \delta(b^{(u)}) &= \delta \begin{pmatrix} c \\ d^{(u)} \end{pmatrix} &= d_{max}^{(u)} - \min(L, d_{min}^{(u)}), \\ \delta(b^{(u+1)}) &= \delta \begin{pmatrix} c \\ d^{(u+1)} \end{pmatrix} &= d_{max}^{(u+1)} - \min(L, d_{min}^{(u+1)}). \end{aligned} \tag{5.8}$$

Now we will compare the differences between the two range norms and there are four cases to consider.

Case (1)

$$\delta(b^{(u)}) = d_{max}^{(u)} - L \text{ and } \delta(b^{(u+1)}) = d_{max}^{(u+1)} - L.$$

From (5.7) we know that $\tau^{(u+1)} = d_{max}^{(u)} - d_{max}^{(u+1)} > 0$. Hence $d_{max}^{(u)} > d_{max}^{(u+1)}$ and therefore $\delta(b^{(u)}) > \delta(b^{(u+1)})$ as required.

Case (2)

$$\delta(b^{(u)}) = d_{max}^{(u)} - d_{min}^{(u)} \text{ and } \delta(b^{(u+1)}) = d_{max}^{(u+1)} - L.$$

By the definition of $x^{(u+1)}$, we know that

$$\begin{aligned} x^{(u)} \geq x^{(u+1)} &\implies & \bar{A}^p \otimes x^{(u)} &\geq \bar{A}^p \otimes x^{(u+1)}, \\ &\implies & d^{(u)} &\geq d^{(u+1)}, \\ &\implies & d_{min}^{(u)} &\geq d_{min}^{(u+1)}. \end{aligned}$$

Therefore $d_{min}^{(u+1)} \geq L \geq d_{min}^{(u)} \geq d_{min}^{(u+1)}$ and hence they are equal. This implies that Case (2) is equivalent to Case (1).

Case (3)

$$\delta(b^{(u)}) = d_{max}^{(u)} - L \text{ and } \delta(b^{(u+1)}) = d_{max}^{(u+1)} - d_{min}^{(u+1)}.$$

Using the definition of $x^{(u+1)}$ again, we know that

$$\begin{aligned}
\tau^{(u+1)} \otimes x^{(u+1)} \geq x^{(u)} \implies \bar{A}^p \otimes (\tau^{(u+1)} \otimes x^{(u+1)}) &\geq \bar{A}^p \otimes x^{(u)}, \\
\implies \tau^{(u+1)} \otimes (\bar{A}^p \otimes x^{(u+1)}) &\geq d^{(u)}, \\
\implies \tau^{(u+1)} \otimes d^{(u+1)} &\geq d^{(u)}, \\
\implies \tau^{(u+1)} + d_{min}^{(u+1)} &\geq d_{min}^{(u)}, \\
\implies d_{max}^{(u)} - d_{max}^{(u+1)} + d_{min}^{(u+1)} &\geq d_{min}^{(u)}, \quad \text{by (5.7)} \\
\implies d_{max}^{(u)} - d_{min}^{(u)} &\geq d_{max}^{(u+1)} - d_{min}^{(u+1)}.
\end{aligned} \tag{5.9}$$

Since $L \leq d_{min}^{(u)}$, therefore

$$\begin{aligned}
\delta(b^{(u)}) &= d_{max}^{(u)} - L, \\
&\geq d_{max}^{(u)} - d_{min}^{(u)}, \\
&\geq d_{max}^{(u+1)} - d_{min}^{(u+1)} \text{ by (5.9)}, \\
&= \delta(b^{(u+1)})
\end{aligned}$$

as required.

Case (4)

$$\Delta(x^{(u)}) = d_{max}^{(u)} - d_{min}^{(u)} \text{ and } \Delta(x^{(u+1)}) = d_{max}^{(u+1)} - d_{min}^{(u+1)}.$$

It immediately follows from (5.9) that $\delta(b^{(u)}) \geq \delta(b^{(u+1)})$. □

Corollary 5.2.1. Let $A \in \overline{\mathbb{R}}^{m \times n}$ be doubly $\mathbb{R}$-astic and $c \in \mathbb{R}^p$ be the vector of prescribed components. Suppose that $d^{(u)} \geq g \geq d^{(u+1)}$ where $d^{(u)}$ and $d^{(u+1)}$ are the d vectors obtained from the u and $u+1$ steps of Algorithm 2, then

$$\delta \begin{pmatrix} c \\ d^{(u)} \end{pmatrix} \geq \delta \begin{pmatrix} c \\ g \end{pmatrix} \geq \delta \begin{pmatrix} c \\ d^{(u+1)} \end{pmatrix}.$$

Proof. It immediately follows from Proposition 5.2.4. $\square$

Proposition 5.2.5. Let $A \in \overline{\mathbb{R}}^{m \times n}$ be doubly $\mathbb{R}$-astic and $c \in \mathbb{R}^p$ be the vector of prescribed components. Suppose that $\hat{d}$ be the vector obtained from Algorithm 2 and $\widetilde{d}$ be an optimal solution to Problem 5, then

$$\delta \begin{pmatrix} c \\ \widetilde{d} \end{pmatrix} = \delta \begin{pmatrix} c \\ \hat{d} \end{pmatrix}.$$

Proof. By Proposition 5.2.4, we know that $\delta \begin{pmatrix} c \\ \hat{d} \end{pmatrix}$ gives the best range norm from all the other vectors produced in the algorithm. Therefore it is the best possible feasible solution produced from Algorithm 2.

Now suppose that $\widetilde{d}$ is an optimal solution to Problem 5 and let $\widetilde{d}_{min}$ and $\widetilde{d}_{max}$ be the minimum and maximum value of $\widetilde{d}$ respectively. We will also let $\hat{d}_{min}$ and $\hat{d}_{max}$ be the minimum and maximum value of $\hat{d}$. There are four cases to consider:

Case (1)

$$\hat{d}_{max} = U \text{ and } \hat{d}_{min} \geq L.$$

In this case, there is nothing to prove since we cannot improve the vector anymore. Therefore $\hat{d}$ is an optimal solution to Problem 5.

Case (2)

$$\hat{d}_{max} = U \text{ and } \hat{d}_{min} < L.$$

Suppose there exists $\widetilde{d}$ such that $\widetilde{d}_{min} > \hat{d}_{min}$ and we will let $\alpha = \widetilde{d}_{min} - \hat{d}_{min} > 0$. Then

$$\begin{aligned}\widetilde{d}_l = \max_{j=1,\ldots,n}(a_{l+p,j} + \widetilde{x}_j) \geq \widetilde{d}_{min} \\ &= \max_{j=1,\ldots,n}(a_{i+p,j} + \widetilde{x}_j) \\ &= \hat{d}_{min} + \alpha \\ &= \max_{j=1,\ldots,n}(a_{l+p,j} + \hat{x}_j) + \alpha.\end{aligned}$$

Let us denote $Q = \{q \in N \mid a_{l+p,q} + \hat{x}_q = \hat{d}_{min}\}$ and $K = \{k \in N \mid a_{r+p,k} + \hat{x}_k = \hat{d}_{max}\}$. Then we will have $\widetilde{x}_q \geq \hat{x}_q + \alpha$, $\forall c \in C$. Since $\widetilde{x} \leq \bar{x}$ and $\hat{x}_j = \bar{x}_j$, $\forall j \notin K$, therefore we have $C \subseteq K$.

Since $Q \subseteq K$, this implies that $\forall q \in Q$ such that

$$\begin{aligned}\hat{d}_{max} = a_{r+p,q} + \hat{x}_q \leq a_{r+p,c} + \widetilde{x}_c - \alpha \\ &\leq \max_{j=1,\ldots,n}(a_{r+p,j} + \widetilde{x}_j) - \alpha \\ &\leq \widetilde{d}_{max} - \alpha.\end{aligned}$$

If $L \geq \widetilde{d}_{min}$, then

$$\begin{aligned}\delta\begin{pmatrix} c \\ \widetilde{d}\end{pmatrix} &= \widetilde{d}_{max} - \widetilde{d}_{min} \\ &\geq \hat{d}_{max} + \alpha - \widetilde{d}_{min} \\ &= \hat{d}_{max} + \alpha - \hat{d}_{min} - \alpha \\ &= \hat{d}_{max} - \hat{d}_{min} \\ &= U - \hat{d}_{min} \\ &= \delta\begin{pmatrix} c \\ \hat{d}\end{pmatrix}.\end{aligned} \tag{5.10}$$

Since $\widetilde{d}$ is an optimal solution to Problem 5, this implies that both sides must be equal and hence $\hat{d}$ is also an optimal solution. On the other hand, suppose that $L < \widetilde{d}_{min}$, then

$$\begin{aligned}
\delta\begin{pmatrix} c \\ \widetilde{d} \end{pmatrix} &= \widetilde{d}_{max} - L \\
&\geq \hat{d}_{max} + \alpha - L \\
&> \hat{d}_{max} + \alpha - \hat{d}_{min} \\
&> \hat{d}_{max} - \hat{d}_{min} \\
&= U - \hat{d}_{min} \\
&= \delta\begin{pmatrix} c \\ \hat{d} \end{pmatrix}
\end{aligned} \tag{5.11}$$

which is a contradiction.

Case (3)

$$\hat{d}_{max} > U \text{ and } \hat{d}_{min} \geq L.$$

Suppose that $\widetilde{d}_{max} < \hat{d}_{max}$ then $\forall k \in K$,

$$\begin{aligned}
\hat{d}_{max} = a_{l+p,k} + \hat{x}_k &> \widetilde{d}_{max} \\
&\geq \widetilde{d}_l \\
&= \max_{j=1,\ldots,n}(a_{l+p,j} + \widetilde{x}_j)
\end{aligned}$$

which implies $\hat{x}_k > \widetilde{x}_k$. But by Algorithm 2, we know that $\widetilde{x}$ is not feasible which is a contradiction. Therefore $\hat{d}$ is an optimal solution to Problem 5.

Case (4)

$$\hat{d}_{max} > U \text{ and } \hat{d}_{min} < L.$$

If there exists $\widetilde{d}_{max} < \hat{d}_{max}$, then immediately by case (3), we know that it is not possible.

Therefore $\widetilde{d}_{max} \geq \hat{d}_{max}$ and $\widetilde{d}_{min} > \hat{d}_{min}$. But this is the same as case (2) and therefore we can conclude that $\hat{d}$ is an optimal solution to Problem 5. □

By the above proposition, we have shown that Algorithm 2 provides us an optimal solution to Problem 5.

5.3 Maximizing the Range Norm

Now we will consider on maximizing the range norm with some components prescribed. We will use the same assumption as before and as a reminder they are:

- The vector of prescribed components is finite and it is an image of the upper matrix, namely A^p, i.e. $c \in Im(A^p) \cap \mathbb{R}^p$.
- The undetermined starting time is also finite.

But this time the objective function will be maximized. Then the problem can be formulated as follows:

Problem 6. Given $A \in \overline{\mathbb{R}}^{m \times n}$ and $c \in Im(A^p) \cap \mathbb{R}^p$, $0 \leq p \leq m$, find $d \in \mathbb{R}^{m-p}$ such that

$$\begin{aligned} \delta(b) \rightarrow & \quad \max \\ \text{subject to} & \quad b \in Im(A) \\ & \quad b^T = (c^T, d^T). \end{aligned}$$

5.3.1 The Case when Only One Machine is Prescribed

We will again consider the case when only one machine is prescribed first, that is $p = 1$. We will show that in this special case, Problem 6 can be considered the same as its counterpart part in Chapter 4, i.e. Problem 4. This can be seem by the following propositions.

Proposition 5.3.1. Let $A \in \mathbb{R}^{m\times n}$ and $c = (c_1) \in \mathbb{R}$ be the prescribed component. Suppose that $b \in Im(A)$ such that b is an optimal solution to Problem 4 and $d = (c_1 \otimes b_1^{-1}) \otimes b$ where b_1 is the first component of b, then the vector $(d_2, ..., d_m)^T$ is an optimal solution to Problem 6.

Proof. Since $d_1 = (c_1 \otimes b_1^{-1}) \otimes b_1 = c_1$, then $(d_2, ..., d_m)^T$ is a feasible solution for Problem 6. We know that b is an optimal solution to Problem 4, therefore any multiple of b is also an optimal solution to Problem 4, i.e. $\delta(b) = \delta(d)$. Since $d \in Im(A)$ and it has the maximum range norm, it immediately follows that $(d_2, ..., d_m)^T$ is an optimal solution to Problem 6. □

Proposition 5.3.2. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and non-finite, i.e. $\exists i \in M$, $j \in N$ such that $a_{ij} = \epsilon$. Suppose that $c = (c_1) \in \mathbb{R}$ is the vector of prescribed component, then Problem 6 is unbounded.

Proof. It immediately follows from Proposition 4.3.2. □

5.3.2 The Case when All but One Machine are Prescribed

Next we will consider the case when $p = m-1$, i.e. the case when all but one of the starting time are prescribed. We will investigate this case by using method similar to that for the minimization problem.

Let $A \in \overline{\mathbb{R}}^{m\times n}$ be a given doubly $\mathbb{R}$-astic matrix and $c \in Im(A^p) \cap \mathbb{R}^p$ be the vector of prescribed components, then $\exists \bar{x} \in \overline{\mathbb{R}}^n$ where $\bar{x} = (A^{m-1})^* \otimes' c$ such that

$$c = A^{m-1} \otimes \bar{x}.$$

Using the vector $\bar{x}$, we will obtain

$$\bar{d} = \bar{A}^{m-1} \otimes \bar{x}.$$

We shall let L and U be the minimum and the maximum value of c respectively, i.e.

$$L = \min_{i=1,\ldots,m-1} c_i,$$

$$U = \max_{i=1,\ldots,m-1} c_i.$$

We will again have three cases to consider:

$$\begin{array}{llll} 1) & L \leq \bar{d} \leq U, & \text{i.e.} & \delta(b) = \delta(c) = U - L \\ 2) & \bar{d} < L, & \text{i.e.} & \delta(b) = U - \bar{d} \\ 3) & U < \bar{d}, & \text{i.e.} & \delta(b) = \bar{d} - L. \end{array}$$

Before we consider each case individually, let us consider the following proposition:

Proposition 5.3.3. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic, $c \in Im(A^p) \cap \mathbb{R}^p$ and $\widetilde{d}$ be an optimal solution to Problem 6. Suppose that $\bar{x} = (A^p)^* \otimes' c$ and $\bar{d} = \bar{A}^p \otimes \bar{x}$, then $\widetilde{d} \leq \bar{d}$.

Proof. Since the proof in Proposition 5.2.3 does not require us to consider the range norm of $\widetilde{d}$, the proof of Proposition 5.3.3 immediately follows from the proof of Proposition 5.2.3.

□

From Proposition 5.3.3, we have again found an upper bound of d for Problem 5 and it is also $\bar{d}$. Therefore we can only decrease the value for $\bar{d}$ in order to improve the range norm of the resulting vector.

Let us look back at the three cases; for the first case, we can see that the range norm can only be improved when the value of $\bar{d}$ is replaced with a smaller value such that it is smaller than L. In fact, in order to obtain an optimal solution for Problem 6, we will need to find a new d such that it is as small as possible. We can also see that this also applies to the second case.

Now we will look at the third case, this case is a bit more complicated than the other cases. This is because when we decrease the value of $\bar{d}$, we will actually obtain a worse range norm than before. But it may happen that we obtain a small enough value such that the range norm of the resulting vector is smaller than the range norm we started with. Therefore we will need to take this into account.

In order to find the smallest value we can replace $\bar{d}$, we will use the method similar as for the minimization problem and find $d^{(1)}$, $d^{(2)}$, etc. Again, we will stop when we obtain a feasible solution $d^{(l)}$, $l \in \mathbb{N}$ such that $d^{(l+1)}$ is not a feasible solution anymore. But then we will need to compare with our first feasible solution since $d^{(l+1)}$ may not be a better feasible solution and the value which is the best range norm out of the two will be our optimal solution to Problem 6.

We also need to note that it may be possible that we find a $\tau^{(q)}$ such that it is equal to $+\infty$, then it immediately implies that we can set $d^{(q)}$ to be arbitrarily small and therefore the optimal solution to Problem 6 is unbounded.

5.3.3 The General Case

Using the observations we have seen above, we can see that the maximization problem can be solved using the same method as the minimization problem. Therefore we are going to modify Algorithm 2 in order to solve Problem 6 for the general case, i.e. $2 \leq p \leq m-1$.

Let us suppose that $A \in \overline{\mathbb{R}}^{m \times n}$ be a given doubly $\mathbb{R}$-astic matrix and $c \in Im(A^p) \cap \mathbb{R}^p$ be the vector of prescribed components, then $\exists \bar{x} \in \overline{\mathbb{R}}^n$ where $\bar{x} = (A^p)^* \otimes' c$ such that $A^p \otimes \bar{x} = c$.

We will let $\bar{d} = \bar{A}^p \otimes \bar{x}$ and $\forall j \in N$,

$$P_j = \{q \in P \mid (a_{qj} - c_q) = \max_{i=1,\ldots,p} (a_{ij} - c_i)\}.$$

We will also let $\bar{d}_{min}$ be the minimum value of $\bar{d}$, i.e.

$$\bar{d}_{min} = \min_{i=1,...,m-p} \bar{d}_i.$$

The range norm may be improved by replacing the corresponding entries in $\bar{d}$ with a smaller value. Therefore we will let

$$R = \{r \in \{1, ..., m-p\} \mid \bar{d}_r = \bar{d}_{min}\}$$

and

$$K = \{k \in N \mid \exists r \in R, \bar{d}_r = \max_{j=1,...,n}(a_{r+p,j} + \bar{x}_j) = a_{r+p,k} + \bar{x}_k\}.$$

We will also set a new x which will be called $x^{(1)}$ such that

$$x_j^{(1)} = \begin{cases} \bar{x}_j - \tau^{(1)} & \text{if } j \in K \\ \bar{x}_j & \text{otherwise} \end{cases}$$

where $\tau^{(1)} > 0$.

The next step will be checking the feasibility of the new value. If

$$\bigcup_{j \notin K} P_j = P.$$

is false, then the new value is not feasible and $\bar{d}$ will be our optimal solution to Problem 6. Otherwise it is feasible and we will need to determine a value for $\tau^{(1)}$. By using the same method as the minimization problem, we can find $\tau^{(1)}$ by (5.7), i.e.

$$\tau^{(1)} = \bar{d}_{max} - \max_{i=p+1,...,m}(\max_{\substack{j=1,...,n \\ j \notin K}}(a_{ij} + \bar{x}_j)).$$

If $\tau^{(1)}$ is equal to $+\infty$, i.e.

$$\max_{i=p+1,\ldots,m} \big(\max_{\substack{j=1,\ldots,n \\ j \notin K}} (a_{ij} + \bar{x}_j) \big) = \epsilon,$$

then it immediately follows that the optimal solution is unbounded.

If this is not the case, then we will repeat the process again and replace $\bar{x}$ and $\bar{d}$ with $x^{(1)}$ and $d^{(1)}$ respectively and find $x^{(2)}$, $d^{(2)}$ and $\tau^{(2)}$, etc. We will stop after we cannot find another feasible solution and finally we will compare the range norm from the vector resulting from the last vector and initial vector $\bar{d}$.

Algorithm 3.
Input: $A \in \overline{\mathbb{R}}^{m \times n}$ and $c \in \mathbb{R}^p$, $2 \leq p \leq m-1$
Output: $\hat{d} \in \mathbb{R}^{m-p}$, an optimal solution to Problem 6 or the optimal solution to Problem 6 is unbounded.

Set $x^{(0)} := (A^p)^* \otimes' c$, $d^{(0)} := \bar{A}^p \otimes x^{(0)}$,

$$d^{(0)}_{min} := \min_{i=1,\ldots,m-p} d^{(0)}_i$$

For all $j \in N$, find

$$P_j = \{q \in P \mid (a_{qj} - c_q) = \max_{i=1,\ldots,p} (a_{ij} - c_i)\}.$$

For $u = 0$ to n do
Begin
Find

$$R = \{r \in \{1, \ldots, m-p\} \mid d^{(u)}_r = d^{(u)}_{min}\}$$

and

$$K = \{k \in N \mid \exists r \in R, \bar{d}_r = \max_{j=1,\ldots,n} (a_{r+p,j} + \bar{x}_j) = a_{r+p,k} + \bar{x}_k\}.$$

If

$$\bigcup_{j \notin K} P_j = P$$

Set $x^{(u+1)}$ by the following:

$$x_j^{(u+1)} := \begin{cases} x_j^{(u)} - \tau^{(u+1)} & \text{if } j \in K, \\ x_j^{(u)} & \text{otherwise} \end{cases}$$

where

$$\tau^{(u+1)} := d_{max}^{(u)} - \max_{i=p+1,\dots,m} \big(\max_{\substack{j=1,\dots,n \\ j \notin K}} (a_{ij} + x_j^{(u)}) \big).$$

If $\tau^{(u+1)} = +\infty$, then the optimal solution is unbounded. Stop.
Set $d^{(u+1)} := \bar{A}^p \otimes x^{(u+1)}$ and

$$d_{max}^{(u+1)} := \max_{i=1,\dots,m-p} d_i^{(u+1)}.$$

Else
if $\delta \begin{pmatrix} c \\ d^{(u)} \end{pmatrix} = \max(\delta \begin{pmatrix} c \\ \bar{d} \end{pmatrix}, \delta \begin{pmatrix} c \\ d^{(u)} \end{pmatrix})$, then $\hat{d} := d^{(u)}$. Stop.
Else $\hat{d} := \bar{d}$. Stop.
End

5.4 Summary

In this chapter, we have investigated the case of minimizing and maximizing range norm of an image set when some of the components of the vector are given and fixed. We have shown that for the case when only one component is prescribed, the minimization and the maximization problem is very similar to the counterpart in Chapter 4. We have also shown that the method for solving the general case of the minimization problem can be modified to solve the maximization case and we have developed algorithms to solve the minimization and maximization problems.

Chapter 6

Integer Linear Systems

6.1 Introduction

In real-life situation, the manufacturers may prefer to have the completion times for their products in terms of hours or minutes, i.e. integer completion time vector. One of the reason of being this is because they may find it easier to keep track of the system this way. But unfortunately the production times for each component in different machines do not necessary last in term of the same units the manufacturers desired, i.e. non-integer production matrix.

Similarly, if we are considering the steady state problem, the fundamental eigenvectors of a production system do not necessarily consist of integer values. But the manufacturers may wish to choose the starting times for their machines as an integer eigenvector due to a similar reason we discussed above. Our goal is to find out if such vectors exist and whether can we find all such vectors.

When we model this question in terms of max-algebra, the problem of finding an integer image vector b from a non-integer production matrix A is called *integer linear system problem* and it can be defined as follows:

Problem 7. Given $A \in \overline{\mathbb{R}}^{m \times n}$, find $b \in \mathbb{Z}^m$ such that

$$A \otimes x = b \text{ where } x \in \overline{\mathbb{R}}^n.$$

This is equivalent to the task of deciding whether if there exists a vector $b \in \mathbb{Z}^m$ such that $b \in Im(A)$.

Since we are only looking for images of a matrix which only consists of integer components, by the argument in Chapter 2 we can assume that A is doubly $\mathbb{R}$-astic for the rest of this chapter. Before we investigate this problem in detail, we will need to define the following notations and definitions.

Definition 6.1.1. If $A \in \overline{\mathbb{R}}^{m \times n}$ then $IIm(A) = Im(A) \cap \mathbb{Z}^m$, i.e. the set of integral images of A.

Definition 6.1.2. Let $\alpha \in \mathbb{R}$, we will denote $f(\alpha) = \alpha - \lfloor \alpha \rfloor$, i.e. $f(\alpha)$ is the fractional part of α.

For completeness, we will denote $f(\epsilon) = \epsilon$ and $\lceil \epsilon \rceil = \epsilon = \lfloor \epsilon \rfloor$.

Definition 6.1.3. Let $A \in \overline{\mathbb{R}}^{m \times n}$, we will denote $\forall j \in N$, $\forall r \in M$,

$$I_j(A, r) = \{i \in M \mid f(a_{ij}) = f(a_{rj})\}.$$

Definition 6.1.4. Let $A \in \mathbb{R}^{m \times n}$, we will define the relation $\sim_j$ such that $\forall j \in N$,

$$a_{pj} \sim_j a_{qj} \;\text{ if }\; p \in I_j(A, q).$$

Now we can see that $\forall j \in N$ and $\forall p, q, s \in M$

$$\begin{array}{llll} \textbf{1)} & a_{pj} \sim_j a_{pj}, & & \\ \textbf{2)} & a_{pj} \sim_j a_{qj} & \Rightarrow a_{qj} \sim a_{pj} & \text{and} \\ \textbf{3)} & a_{pj} \sim_j a_{qj},\ a_{qj} \sim_j a_{sj} & \Rightarrow a_{pj} \sim_j a_{sj}. & \end{array}$$

Therefore $\sim_j$ defines an equivalence relation and $I_j(A, r)$ are the equivalence classes.

We will also need the following definition for max-algebraic linear system which were presented in Chapter 2. Suppose we have the system $A \otimes x = b$ where $A = (a_{ij}) \in \overline{\mathbb{R}}^{m \times n}$ and $b = (b_1, ..., b_m) \in \overline{\mathbb{R}}^m$ are given, then we denote

$$S(A, b) = \{x \in \overline{\mathbb{R}}^n \mid A \otimes x = b\},$$

$$M_j(A, b) = \{k \in M \mid (a_{kj} - b_k) = \max_{i=1,...,m}(a_{ij} - b_i)\}, \quad \forall j \in N,$$

$$\bar{x}_j = -\max_{i=1,...,m}(a_{ij} - b_i), \quad \forall j \in N.$$

In order to find out if integral images exist for any matrix A, we will first need to determine some necessary conditions.

Proposition 6.1.1. Let $A \in \overline{\mathbb{R}}^{m \times n}$ be doubly $\mathbb{R}$-astic and $b \in IIm(A)$, then $\forall j \in N$, $\exists r_j \in M$ such that

$$M_j(A, b) \subseteq I_j(A, r_j).$$

Proof. Let $A \in \overline{\mathbb{R}}^{m \times n}$ be doubly $\mathbb{R}$-astic and $b \in IIm(A)$. Suppose that $j \in N$ and $k \in M_j(A, b)$, then $\forall i \in M_j(A, b)$

$$a_{kj} - b_k = a_{ij} - b_i,$$

$$f(a_{kj} - b_k) = f(a_{ij} - b_i),$$

$$(a_{kj} - b_k) - \lfloor (a_{kj} - b_k) \rfloor = (a_{ij} - b_i) - \lfloor (a_{ij} - b_i) \rfloor.$$

Since $b \in IIm(A)$, we know that $b_l \in \mathbb{Z}, \forall l$ and it implies

$$\lfloor (a_{lj} - b_l) \rfloor = \lfloor a_{lj} \rfloor - b_l.$$

Therefore we have

$$(a_{kj} - b_k) - (\lfloor a_{kj} \rfloor - b_k) = (a_{ij} - b_i) - (\lfloor a_{ij} \rfloor - b_i),$$
$$a_{kj} - \lfloor a_{kj} \rfloor = a_{ij} - \lfloor a_{ij} \rfloor,$$
$$f(a_{kj}) = f(a_{ij})$$

and hence $\forall i \in M_j(A, b)$, $i \in I_j(A, k)$. $\square$

Corollary 6.1.1. Let $A \in \overline{\mathbb{R}}^{m \times n}$ be doubly $\mathbb{R}$-astic and $b \in IIm(A)$. Suppose that $J \subseteq N$ such that

$$\bigcup_{j \in J} M_j(A, b) = M$$

then $\forall j \in J$, $\exists r_j \in M$ such that

$$\bigcup_{j \in J} I_j(A, r_j) = M.$$

Proof. It immediately follows from Proposition 6.1.1. $\square$

6.2 The Case of One Column Matrix

Now we will investigate the integer linear system problem. We will start by considering the easier cases and we will move on to the general cases in the latter part of this chapter. First, we will consider the case when the matrix A has only one column, i.e. $n = 1$.

Using Corollary 6.1.1, we can obtain the following proposition which identifies the existence of an integer image for any one column matrix.

Proposition 6.2.1. Let $A \in \overline{\mathbb{R}}^{m\times 1}$ and $A \neq \epsilon$, $IIm(A) \neq \emptyset$ if and only if $\exists r \in M$ such that $I_1(A, r) = M$, i.e. all entries in column one have the same fractional part.

Proof. "$\Rightarrow$" immediately follows from Corollary 6.1.1. Now let us suppose that $\exists r \in M$ such that $I_1(A, r) = M$. It implies that every element in A must have the same fractional part.

Now if we choose a $x = (x_1)$ such that $f(a_{i1} + x_1) = 0$, $\forall i \in M$, then we can see that $A \otimes x \in \mathbb{Z}^m$. $\square$

Hence Proposition 6.2.1 provides a necessary and sufficient condition for the case when $n = 1$. We can also use a similar idea to solve a special case for $A \in \overline{\mathbb{R}}^{m\times n}$.

Proposition 6.2.2. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic, if $\exists k \in N$, $\exists r \in M$ such that $I_k(A, r) = M$ then $IIm(A) \neq \emptyset$.

Proof. Suppose that $\exists k \in N$, $\exists r \in M$ such that $I_k(A, r) = M$. We can find a $x \in \overline{\mathbb{R}}^n$ by setting $x_k \in \mathbb{R}$ such that $f(a_{ik} + x_k) = 0$, $\forall i \in M$ and $x_j = \epsilon$, $\forall j \neq k$. By doing this, we will obtain the vector $b = A \otimes x$ and $b \in \mathbb{Z}^m$. $\square$

Proposition 6.2.2 gives a sufficient condition for $IIm(A) \neq \emptyset$ for the general case. The next step will be to consider what happens if this condition is not met. The aim is to find a both necessary and sufficient for the general case. Therefore we will suppose that the condition of Proposition 6.2.1 is not satisfied for the rest of this chapter.

6.3 The Case of Two Columns Matrix

Next we will consider the case when the matrix only consists of two columns, i.e. $n = 2$. Now the integer linear system problem is less straight forward than the one column case. First let us consider the following proposition.

Proposition 6.3.1. [23][62] Let $A \in \overline{\mathbb{R}}^{m\times 2}$ be doubly $\mathbb{R}$-astic, $m > 1$, then there exist $s, t \in M$, $s \neq t$ such that $s \in M_1(A, b)$, $t \in M_2(A, b)$ for any $b \in Im(A)$. Furthermore, s and t are the indices such that

$$a_{s1} - a_{s2} = \max_{i=1,\ldots,m} (a_{i1} - a_{i2}),$$

$$a_{t1} - a_{t2} = \min_{i=1,\ldots,m} (a_{i1} - a_{i2}).$$

Proof. Let $b \in Im(A)$, then $\forall i \in M$ we have either $i \in M_1(A, b)$ or $i \in M_2(A, b)$. Suppose that $s \notin M_1(A, b)$ then $\exists r \in M$ such that

$$a_{r1} - b_r > a_{s_1} - b_s. \tag{6.1}$$

At the same time, we know that $b \in Im(A)$ and therefore $s \in M_2(A, b)$ and $\forall i \in M$,

$$a_{s2} - b_s \geq a_{i2} - b_i$$

and hence

$$a_{s2} - b_s \geq a_{r2} - b_r. \tag{6.2}$$

By adding (6.1) and (6.1), we get

$$a_{r1} - b_r + a_{s2} - b_s > a_{s_1} - b_s + a_{r2} - b_r,$$

$$a_{r1} + a_{s2} > a_{s_1} + a_{r2},$$

$$a_{r1} - a_{r2} > a_{s1} - a_{s2}$$

which is a contradiction with the choice of s. Hence $r \in M_1(A, b)$. It is proved similarly that $s \in M_2(A, b)$. $\square$

By the above proposition, we know that $\forall b \in Im(A)$, $s \in M_1$ and $t \in M_2$. Let us suppose that $s = t$, then it immediately implies

$$\max_{i=1,\dots,m}(a_{i1} - a_{i2}) = \min_{i=1,\dots,m}(a_{i1} - a_{i2}).$$

But this can only happen when $(a_{i2} - a_{i1}) = \alpha$, $\forall i \in M$ where $\alpha \in \mathbb{R}$. This means that the second column of A is a multiple of the first column of A and this is equivalent to the case when $n = 1$. Therefore without loss of generality, we can assume $s \neq t$ for the rest of this section.

Now using Proposition 6.3.1, we can obtain the following:

Proposition 6.3.2. Let $A \in \overline{\mathbb{R}}^{m\times 2}$ be doubly $\mathbb{R}$-astic and

$$a_{s1} - a_{s2} = \max_{i=1,\dots,m}(a_{i1} - a_{i2}),$$

$$a_{t1} - a_{t2} = \min_{i=1,\dots,m}(a_{i1} - a_{i2}),$$

then $IIm(A) \neq \emptyset$ if and only if $\exists M_1$, M_2 such that

$$1)\ \ M_1 \bigcup M_2 = M \text{ and}$$

$$2)\ \ \exists \beta \in \mathbb{Z} \text{ such that}$$

$$\lfloor (a_{t2} - a_{s1}) + \min_{i\in M_1}(a_{i1} - a_{i2}) \rfloor \geq \beta \geq \lceil (a_{t2} - a_{s1}) + \max_{i\in M_2}(a_{i1} - a_{i2}) \rceil.$$

Furthermore, we can generate all vectors in $IIm(A)$ by taking all values of β satisfying 2) and replace it in the following:

$$b_r = \mu + a_{r1} - a_{s1},$$

$$b_q = \mu + a_{q2} - a_{t2} + \beta,$$

$\forall r \in M_1$, $q \in M_2$ and $\mu \in \mathbb{Z}$.

Proof. " $\Rightarrow$ " Suppose that $b = (b_1, b_2, ..., b_m)^T \in IIm(A)$ then by Corollary 2.3.1, we have $M_1(A, b) \bigcup M_2(A, b) = M$.

Since we can permute any rows in the matrix, i.e. renumbering the rows and the problem will not be changed. Therefore without loss of generality, we can assume $s = 1$ and $t = 2$. Also any integer multiple of b will still be an integer image of A, therefore we will multiply b by b_1^{-1} and $\forall i \in M$, we can relabel $b_i \otimes b_1^{-1} = b_i$ and hence we will have $b_1 = 0$.

By Proposition 6.3.1, we have $1 \in M_1(A, b)$ and therefore

$$\begin{aligned} a_{11} - b_1 = a_{11} &= \max_{i=1,\dots,m} (a_{i1} - b_i), \\ &= a_{r1} - b_r, \quad \forall r \in M_1(A, b), \\ b_r &= a_{r1} - a_{11}, \quad \forall r \in M_1(A, b) \end{aligned} \tag{6.3}$$

and

$$a_{11} \geq a_{q1} - b_q \;\; \forall q \in M_2(A, b). \tag{6.4}$$

Similarly, we have $2 \in M_2(A, b)$ and therefore we have

$$\begin{aligned} a_{22} - b_2 &= \max_{i=1,\dots,m} (a_{i2} - b_i), \\ &= a_{q2} - b_q, \quad \forall q \in M_2(A, b), \\ b_q &= a_{q2} - a_{22} + b_2, \quad \forall q \in M_2(A, b) \end{aligned} \tag{6.5}$$

and

$$a_{22} - b_2 \geq a_{r2} - b_r, \quad \forall r \in M_1(A, b). \tag{6.6}$$

From (6.3) and (6.6) we have $\forall r \in M_1(A, b)$,

$$\begin{aligned} a_{22} - b_2 &\geq a_{r2} - (a_{r1} - a_{11}), \\ (a_{22} - a_{11}) + (a_{r1} - a_{r2}) &\geq b_2, \\ (a_{22} - a_{11}) + \min_{i \in M_1(A,b)} (a_{i1} - a_{i2}) &\geq b_2. \end{aligned} \tag{6.7}$$

Note that $b \in IIm(A)$, therefore $b_2 \in \mathbb{Z}$. So we can take the largest integer value less than the value on the LHS and we will have

$$\lfloor (a_{22} - a_{11}) + \min_{i \in M_1(A,b)} (a_{i1} - a_{i2}) \rfloor \geq b_2. \tag{6.8}$$

Now if we consider (6.4) and (6.5), we have $\forall q \in M_2(A, b)$,

$$\begin{aligned} a_{11} &> a_{q1} - (a_{q2} - a_{22} + b_2), \\ b_2 &> (a_{q1} - a_{q2}) + (a_{22} - a_{11}), \\ b_2 &> (a_{22} - a_{11}) + \max_{i \in M_2(A,b)} (a_{i1} - a_{i2}). \end{aligned} \tag{6.9}$$

By a similar argument as before, we can take the smallest integer value greater than the value on the RHS and we will have

$$b_2 \geq \lceil (a_{22} - a_{11}) + \max_{i \in M_2(A,b)} (a_{i1} - a_{i2}) \rceil. \tag{6.10}$$

Combining the two inequalities (6.8) and (6.10), then we have found the following bounds for b_2:

$$\lfloor (a_{22} - a_{11}) + \min_{i \in M_1(A,b)} (a_{i1} - a_{i2}) \rfloor \geq b_2 \geq \lceil (a_{22} - a_{11}) + \max_{i \in M_2(A,b)} (a_{i1} - a_{i2}) \rceil. \tag{6.11}$$

" $\Leftarrow$ " Suppose that we have found the sets M_1 and M_2 such that $M_1 \bigcup M_2 = M$ and there exists an integer value β which satisfies (6.11). Then we will let $x_1 = -a_{11}$ and $x_2 = -(a_{22} - \beta)$ and therefore we have

$$b_i = \max\{(a_{i1} - a_{11}), (a_{i2} - a_{22} + \beta)\}.$$

Let $i \in M_1$ and suppose that $a_{i2} - a_{22} + \beta > a_{i1} - a_{11}$, then we have

$$\begin{aligned} a_{i2} - a_{22} + \beta &> a_{i1} - a_{11}, \\ \beta &> (a_{22} - a_{11}) + (a_{i1} - a_{i2}). \end{aligned}$$

But this contradicts (6.11) and therefore

$$a_{i2} - a_{22} + \beta \leq a_{i1} - a_{11}.$$

Hence $\forall i \in M_1$,

$$b_i = a_{i1} - a_{11}.$$

Since $1 \in M_1 \subseteq I_1(A, i)$, we have $\forall i \in M_1$,

$$f(a_{i1}) - f(a_{11}) = 0$$

and hence $b_i \in \mathbb{Z}$.

Using a similarly argument, we can conclude that

$$b_j = a_{j2} - a_{22} + \beta$$

and $\forall j \in M_2$,

$$f(a_{j2}) - f(a_{22}) = 0.$$

Since $\beta \in \mathbb{Z}$, we have $b_j \in \mathbb{Z}$ and therefore $b \in IIm(A)$ and any multiple of b is also in $IIm(A)$. Hence $IIm(A) \neq \emptyset$. $\square$

The above proposition provides a necessary and sufficient condition for the existence of a solution in the case when $n = 2$. But the difficult part is to find the sets M_1 and M_2 such that (6.11) is satisfied.

By Corollary 6.1.1, we know that if $b \in IIm(A)$ exists then $\forall j \in J \subseteq N$, $\exists r_j \in M$ such that

$$\bigcup_{j \in J} I_j(A, r_j) = M.$$

Hence we know there exist $r_1, r_2 \in M$ such that $I_1(A, r_1) \cup I_2(A, r_2) = M$ for the case when $n = 2$. The next step will be to find r_1 and r_2.

Now using Proposition 6.3.1 and 6.1.1, we have $\forall b \in IIm(A)$,

$$\begin{aligned} & s \in \ M_1(A, b) \subseteq I_1(A, r_1) \\ \Rightarrow \quad & s \in \ I_1(A, r_1) = I_1(A, s) \text{ since they are equivalent classes.} \end{aligned}$$

Similarly, we also have $t \in I_2(A, t)$ and therefore $\forall b \in IIm(A)$ we have

$$I_1(A, s) \cup I_2(A, t) = M. \tag{6.12}$$

Using the above result and Proposition 6.3.2, we can produce the following sufficient condition.

Proposition 6.3.3. Let $A \in \overline{\mathbb{R}}^{m \times 2}$ be doubly $\mathbb{R}$-astic and

$$a_{s1} - a_{s2} = \max_{i=1,\dots,m} (a_{i1} - a_{i2}),$$

$$a_{t1} - a_{t2} = \min_{i=1,\dots,m} (a_{i1} - a_{i2}),$$

then $IIm(A) \neq \emptyset$ if

$$1) \ \ I_1(A,s) \cup I_2(A,t) = M \text{ and}$$

$$2) \ \ \exists \beta \in \mathbb{Z} \text{ such that}$$

$$\lfloor (a_{t2} - a_{s1}) + \min_{i \in I_1(A,s)} (a_{i1} - a_{i2}) \rfloor \geq \beta \geq \lceil (a_{t2} - a_{s1}) + \max_{i \in I_2(A,t)} (a_{i1} - a_{i2}) \rceil.$$

Proof. It immediately follows from Corollary 6.1.1 and Proposition 6.3.2. □

Note that the intersection of the two sets $I_1(A,s)$ and $I_2(A,t)$ may not be empty. The conditions stated above assume that

$$I_1(A,s) \cap I_2(A,t) \subseteq M_1(A,b)$$

and

$$I_1(A,s) \cap I_2(A,t) \subseteq M_2(A,b)$$

which may not happen in general. Therefore the condition above is only a sufficient but not necessary condition for the existence of integer image.

Using this observation, we will generate the following propositions.

Proposition 6.3.4. Let $A \in \overline{\mathbb{R}}^{m \times 2}$ be doubly $\mathbb{R}$-astic and

$$a_{s1} - a_{s2} = \max_{i=1,\ldots,m} (a_{i1} - a_{i2}),$$

$$a_{t1} - a_{t2} = \min_{i=1,\ldots,m} (a_{i1} - a_{i2}).$$

Let

$$\hat{I}_1(A,s) = I_1(A,s) - I_2(A,t) \text{ and } \hat{I}_2(A,t) = I_2(A,t) - I_1(A,s),$$

then $\forall b \in IIm(A)$,

$$\hat{I}_1(A,s) \subseteq M_1(A,b) \text{ and } \hat{I}_2(A,t) \subseteq M_2(A,b).$$

In addition, if

$$I_1(A,s) \cap I_2(A,t) = \emptyset$$

then $I_1(A,s) = M_1(A,b)$ and $I_2(A,t) = M_2(A,b)$.

Proof. Without loss of generality, we can assume that $s = 1$ and $t = 2$. Let $i \in \hat{I}_1(A,1)$ then $i \notin I_2(A,2)$. By Proposition 6.1.1, this implies that $\forall b \in IIm(A)$, $i \notin M_2(A,b)$.

Therefore $i \in M_1(A,b)$ and we have $\hat{I}_1(A,1) \subseteq M_1(A,b)$. By the same argument we will obtain the conclusion that $\hat{I}_2(A,2) \subseteq M_2(A,b)$.

Now Suppose that

$$I_1(A,1) \cap I_2(A,2) = \emptyset,$$

then

$$M_1(A,b) \subseteq I_1(A,1) = \hat{I}_1(A,1) \subseteq M_1(A,b).$$

Hence $\forall b \in IIm(A)$, $I_1(A,1) = M_1(A,b)$ and similarly we have $I_2(A,2) = M_2(A,b)$. $\square$

Now if we use Proposition 6.3.2 with the above result, we will obtain the following proposition.

Proposition 6.3.5. Let $A \in \overline{\mathbb{R}}^{m\times 2}$,

$$a_{s1} - a_{s2} = \max_{i=1,\ldots,m} (a_{i1} - a_{i2}),$$

$$a_{t1} - a_{t2} = \min_{i=1,\ldots,m} (a_{i1} - a_{i2}).$$

Suppose that $I_1(A,s) \cap I_2(A,t) = \emptyset$, then $IIm(A) \neq \emptyset$ if and only if

1) $I_1(A,s) \cup I_2(A,t) = M$ and

2) $\exists \beta \in \mathbb{Z}$ such that

$$\lfloor (a_{t2} - a_{s1}) + \min_{i \in I_1(A,s)} (a_{i1} - a_{i2}) \rfloor \geq \beta \geq \lceil (a_{t2} - a_{s1}) + \max_{i \in I_2(A,t)} (a_{i1} - a_{i2}) \rceil.$$

Furthermore, we can generate all vectors in $IIm(A)$ by taking all values of β satisfying $2)$ and replace it in the following:

$$b_r = \mu + a_{r1} - a_{s1}$$

$$b_q = \mu + a_{q2} - a_{t2} + \beta$$

$\forall r \in I_1(A,s)$, $q \in I_2(A,t)$ and $\mu \in \mathbb{Z}$.

Proof. It immediately follows from Proposition 6.3.2 and 6.3.4. $\square$

Example 6.3.1. Let A be the following matrix:

$$\begin{pmatrix} 0.5 & -5.7 \\ 6.2 & -2.8 \\ 1 & 3.3 \\ 3.5 & -3.7 \\ 9.2 & -1.4 \\ -2.4 & -1.7 \end{pmatrix}$$

Then

$$\begin{array}{rcrcrcr}
a_{11} - a_{12} & = & 0.5 & - & (-5.7) & = & 6.2 \\
a_{21} - a_{22} & = & 6.2 & - & (-2.8) & = & 9 \\
a_{31} - a_{32} & = & 1 & - & 3.3 & = & -2.3 \\
a_{41} - a_{42} & = & 3.5 & - & (-3.7) & = & 7.2 \\
a_{51} - a_{52} & = & 9.2 & - & (-1.4) & = & 10.6 \\
a_{61} - a_{62} & = & -2.4 & - & (-1.7) & = & -0.7
\end{array}$$

and hence

$$\max_{i=1,\ldots,m}(a_{i1} - a_{i2}) = a_{51} - a_{52}$$

$$\min_{i=1,\ldots,m}(a_{i1} - a_{i2}) = a_{31} - a_{32}.$$

Therefore $s = 5$ and $t = 3$. Next we need to find $I_1(A, 5)$ and $I_2(A, 3)$ and we can see that $I_1(A, 5) = \{2, 5\}$ and $I_2(A, 3) = \{1, 3, 4, 6\}$. The two sets are disjoint and therefore we can apply Proposition 6.3.5.

The upper limit for β is

$$\begin{aligned}
\lfloor (a_{t2} - a_{s1}) + \min_{i \in I_1(A,s)}(a_{i1} - a_{i2}) \rfloor &= \lfloor (a_{32} - a_{51}) + \min_{i \in I_1(A,5)}(a_{i1} - a_{i2}) \rfloor \\
&= \lfloor (3.3 - 9.2) + \min(6.2 - (-2.8), 9.2 - (-1.4)) \rfloor \\
&= \lfloor -5.9 + 9 \rfloor \\
&= \lfloor 3.1 \rfloor \\
&= 3.
\end{aligned}$$

And the lower limit is

$$
\begin{aligned}
\lceil (a_{t2} - a_{s1}) + \max_{i \in I_2(A,t)} (a_{i1} - a_{i2}) \rceil &= \lceil (a_{32} - a_{51}) + \max_{i \in I_2(A,5)} (a_{i1} - a_{i2}) \rceil \\
&= \lceil -5.9 + \max(6.2, -2.3, 7.2, -0.7) \rceil \\
&= \lceil -5.9 + 7.2 \rceil \\
&= \lceil 1.3 \rceil \\
&= 2.
\end{aligned}
$$

Hence an integer value for β exists, i.e. $\beta = 2, 3$ and therefore $IIm(A) \neq \emptyset$. In particular the following vectors:

$$
\begin{pmatrix} -7 \\ -3 \\ 2 \\ -5 \\ 0 \\ -3 \end{pmatrix} \text{ and } \begin{pmatrix} -6 \\ -3 \\ 3 \\ -4 \\ 0 \\ -2 \end{pmatrix}
$$

and their integer multiples are the integer images of A.

6.4 Strongly Regular Matrix

In this section, we will investigate the case of strongly regular matrix. The definition of strongly regular matrix can be found in Chapter 2 and comprehensive results regarding this topic can be found in [14]. Also in [14], we can see that there is a relation between the simple image set and the eigenproblem. It turns out that the existence of integer image set and integer simple image set is also related to the eigenproblem which can be seen in the following section.

6.4.1 Basic Principle

Let us consider the case when a $n \times n$ matrix is given. Our aim is to find out the conditions which the matrices have to satisfy in order to have integral images. First let us consider the following proposition.

Proposition 6.4.1. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be doubly $\mathbb{R}$-astic and $\forall j \in N$, $|I_j(A,r)| = 1$, $\forall r \in N$. If $IIm(A) \neq \emptyset$ then A is a strongly regular matrix.

Proof. By Proposition 6.1.1, we know that $\forall b \in IIm(A)$, $\exists r_j \in M$ such that $M_j(A,b) \subseteq I_j(A,r_j)$, $\forall j \in N$. Since $|I_j(A,r)| = 1$, $\forall r$, therefore we have

$$M_j(A,b) = I_j(A,r_j) = \{r_j\},\ \forall j \in N.$$

From Corollary 2.3.1, we know that

$$\bigcup_{j\in N} M_j(A,b) = M = N.$$

Since for any $N' \subseteq N,\ N' \neq N$

$$|\bigcup_{j\in N'} M_j(A,b)| < n = |N|$$

and therefore we also know that for any $N' \subseteq N,\ N' \neq N$

$$\bigcup_{j\in N'} M_j(A,b) \neq N.$$

Hence by Corollary 2.3.2, the system $A \otimes x = b$ has a unique solution. Since $m = n$, A is a strongly regular matrix. $\square$

Corollary 6.4.1. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic, $m \leq n$ and $\forall j \in N$, $|I_j(A,r)| = 1$, $\forall r \in M$. If $IIm(A) \neq \emptyset$, then A contains a strongly regular $m \times m$ sub-matrix.

Proof. Let $b \in IIm(A)$ and by the same argument as before, we have

$$M_j(A,b) = I_j(A,r_j) = \{r_j\},\ \forall j \in N.$$

Since $|M_j(A,b)| = 1$, $\forall j$, then we can construct $N' \subseteq N$ by the following procedure:

Let $M' := N' := \emptyset$.

for $j = 1$ to n do

begin

  if $M_j(A,b) \subseteq M'$

    Set $N' := N' \cup \{j\}$.

    Set $M' := M' \cup M_j(A,b)$

end.

Then M' will be a minimum set covering of M and we can see that $|N'| = |M'| = m$. Now for each element in N', i.e. $j \in N$, we will take the corresponding column from A, i.e. A_j and form a new matrix $\bar{A}$. Then $\bar{A}$ is a square sub-matrix of A and since

$$\bigcup_{j\in N'} M_j(A,b) = \bigcup_{j\in M} M_j(\bar{A},b) = M,$$

we have $b \in IIm(\bar{A})$ and hence $IIm(\bar{A}) \neq \emptyset$. Then it immediately follows from Proposition 6.4.1 that $\bar{A}$ is strongly regular. $\square$

Definition 6.4.1. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic, then A will be called *typical* if $\forall j \in N$, $|I_j(A,r)| = 1$, $\forall r \in M$, i.e. no two entries in A have the same fractional part.

Corollary 6.4.2. If A is typical, then $IIm(A) \subseteq S_A$ where S_A is the simple image set of A. Hence a typical matrix A is strongly regular if it has an integer image.

Proof. It immediately follows from Proposition 6.4.1. $\square$

Note that the inclusion $S_A \subseteq IIm(A)$ is not true in general. This can be seen by the following example.

Example 6.4.1. Let

$$A = \begin{pmatrix} 0 & 0.61 & -1.5 \\ -1.62 & 0.1 & -2.2 \\ -1.16 & -1.34 & 0.3 \end{pmatrix},$$

we can see that no two entries in A have the same fractional part and therefore A is typical. We can also see that the identity is the unique optimal permutation for A and hence A is strongly regular.

Let $b = (0, -0.6, 1.5)^T$, then

$$a_{ij} - b_i = \begin{pmatrix} 0 - 0 & 0.61 - 0 & -1.5 - 0 \\ -1.62 - (-0.6) & 0.1 - (-0.6) & -2.2 - (-0.6) \\ -1.16 - 1.5 & -1.34 - 1.5 & 0.3 - 1.5 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0.61 & -1.5 \\ -1.02 & 0.7 & -1.6 \\ -2.66 & -2.84 & -1.2 \end{pmatrix}.$$

We can see that the diagonal entries are strictly greater than all other entries in the same column. Therefore $b \in S_A$. Since b is not an integer vector, therefore $S_A \nsubseteq IIm(A)$.

We will also see from the following example, that if A is not typical and it has integer image, then it does not imply that A is strongly regular.

Example 6.4.2. Let

$$A = \begin{pmatrix} 0 & 2 & -2.3 \\ -1 & 1 & -1.4 \\ 0.6 & -1.2 & 0.3 \end{pmatrix},$$

we can see that the entries a_{11} and a_{12} are both integers, therefore A is not a typical matrix. It is not difficult to see that there are two optimal permutations for A, i.e. the identity and the permutation $(1\ 2)(3)$. Therefore A is not strongly regular.
Let $x = (0, -2, -0.3)^T$, then

$$\begin{aligned} A \otimes x &= \begin{pmatrix} 0 & 2 & -2.3 \\ -1 & 1 & -1.4 \\ 0.6 & -1.2 & 0.3 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ -2 \\ -0.3 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}. \end{aligned}$$

Therefore $b \in IIm(A)$ and hence $IIm(A) \neq \emptyset$.

Also it turns out that for every strongly regular matrix, it immediately follows that it is doubly $\mathbb{R}$-astic. This property can be seen by the following lemma.

Lemma 6.4.1. Let $A \in \overline{\mathbb{R}}^{n \times n}$ be strongly regular, then A is also doubly $\mathbb{R}$-astic.

Proof. Let A be strongly regular, then $\exists b \in \mathbb{R}^m$ such that the system $A \otimes x = b$ has a unique solution. Suppose that A is not row $\mathbb{R}$-astic, then $\exists r \in M$ such that $\forall j \in N$, $a_{rj} = \epsilon$. But it immediately implies that the system has no solution and it is a contradiction.
Suppose that A is not column $\mathbb{R}$-astic, then $\exists s \in N$ such that $\forall i \in M$, $a_{is} = \epsilon$. Then we can set x_s to be any value in a solution x and it contradicts the fact that the system has a unique solution. Therefore A is doubly $\mathbb{R}$-astic. $\square$

By the next proposition (Proposition 6.4.2), we will show that for any square matrices, if we subtract from each column of the matrix by any constants the image set will remain the same.

Proposition 6.4.2. Let $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$ be doubly $\mathbb{R}$-astic, $k \in \mathbb{R}^n$ and $\hat{A} = (\hat{a}_{ij})$ where $\hat{a}_{ij} = a_{ij} + k_j, \forall i, j \in N$. Then $Im(A) = Im(\hat{A})$.

Proof. Suppose $b \in Im(A)$ then $\exists x \in \overline{\mathbb{R}}^n$ such that

$$
\begin{aligned}
& A \otimes x = b \\
\Longleftrightarrow & \max_{j=1,\dots,n} (a_{ij} + x_j) = b_i, \quad \forall i \in N \\
\Longleftrightarrow & \max_{j=1,\dots,n} (a_{ij} - k_j + k_j + x_j) = b_i, \quad \forall i \in N \\
\Longleftrightarrow & \max_{j=1,\dots,n} (\hat{a}_{ij} + y_j) = b_i, \quad \forall i \in N \text{ where } y_j = k_j + x_j \\
\Longleftrightarrow & \hat{A} \otimes y = b.
\end{aligned}
$$

Therefore $b \in Im(\hat{A})$ and hence $Im(A) = Im(\hat{A})$. □

Similarly we will show (by Proposition 6.4.3) that for any strongly regular matrices, if we subtract the columns by any constants, the simple image set of the matrix will remain unchanged.

Proposition 6.4.3. Let $A = (a_{ij}) \in \overline{\mathbb{R}}^{n\times n}$ be strongly regular and $\pi \in ap(A)$. Suppose that $k \in \mathbb{R}^n$ and $\hat{A} = (\hat{a}_{ij})$ where $\hat{a}_{ij} = a_{ij} - k_j, \forall i, j \in N$. Then $S_A = S_{\hat{A}}$.

Proof. Suppose $b \in S_A$ then $\forall j \in N$,

$$
\begin{aligned}
& a_{j,\pi(j)} - b_j &>& \; a_{i,\pi(j)} - b_i, & \forall i \in N, \\
\Longleftrightarrow \quad & a_{j,\pi(j)} - k_{\pi(j)} - b_j &>& \; a_{i,\pi(j)} - k_{\pi(j)} - b_i, & \forall i \in N, \\
\Longleftrightarrow \quad & \hat{a}_{j,\pi(j)} - b_j &>& \; \hat{a}_{i,\pi(j)} - b_i, & \forall i \in N.
\end{aligned}
$$

Therefore $b \in S_{\hat{A}}$ and hence $S_A = S_{\hat{A}}$. $\square$

Proposition 6.4.4. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be doubly $\mathbb{R}$-astic and $P \in \overline{\mathbb{R}}^{n\times n}$ be a generalized permutation matrix. Then $Im(A \otimes P) = Im(A)$. Furthermore if A is strongly regular, then $S_{(A\otimes P)} = S_A$.

Proof. Suppose $b \in Im(A)$ then $\exists x \in \overline{\mathbb{R}}^n$ such that $A \otimes x = b$. By Theorem 2.2.1, we know that the inverse of P exists. If we let $y = P^{-1} \otimes x$, then

$$A \otimes x = b \Longrightarrow A \otimes P \otimes y = b.$$

Therefore $b \in Im(A \otimes P)$. Similarly if $b \in Im(A \otimes P)$, then $\exists x \in \overline{\mathbb{R}}^n$ such that

$$(A \otimes P) \otimes x = b \Longrightarrow A \otimes y = b$$

where $y = P \otimes x$ and hence $Im(A) = Im(A \otimes P)$.

If A is strongly regular, then $\forall b \in S_A$, $\exists! x \in \overline{\mathbb{R}}^n$ such that $A \otimes x = b$. Therefore

$$\begin{aligned} &\text{The system } A \otimes x = b \text{ has a unique solution} \\ \Longrightarrow &\text{The system } A \otimes P \otimes y = b \text{ has a unique solution.} \end{aligned}$$

Therefore $b \in S_{(A\otimes P)}$. If $b \in S_{(A\otimes P)}$, then $\exists! x \in \overline{\mathbb{R}}^n$ such that

$$\begin{aligned} &\text{The system } (A \otimes P) \otimes x = b \text{ has a unique solution.} \\ \Longrightarrow &\text{The system } A \otimes y = b \text{ has a unique solution} \end{aligned}$$

where $y = P \otimes x$ and hence $S_{(A\otimes P)} = S_A$. $\square$

Since A is multiplied by P from the right, it implies that the columns of A are permuted during this multiplication. Therefore by Proposition 6.4.4, we can without loss of generality, permute the columns of the matrix with any permutation π and the (simple) image set of the resulting matrix will not be affected.

Definition 6.4.2. Let $A \in \overline{\mathbb{R}}^{n\times n}$, then A will be called *strongly definite* if $\forall i \in N$, $a_{ii} = 0$ and $id \in ap(A)$, i.e. all the diagonal entries are zero and the identity is an optimal permutation of A.

Note that for any strongly definite matrix A, its maximum cycle mean must be equal to 0. This can be seen by the following proposition.

Proposition 6.4.5. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly definite, then $\Lambda(A) = \{0\}$. Furthermore $V(A) = Im(\Gamma(A))$ and

$$V^{+}(A) = \{\Gamma(A) \otimes x \mid x \in \mathbb{R}^n\}.$$

Proof. Since the identity is an optimal permutation of A by definition, this immediately implies that the matrix contains no positive cycle. Hence fore, the maximum cycle mean must be equal to 0.

Using the fact that for all strongly regular matrices, $a_{ii} = 0$, $\forall i \in N$ this immediately implies that all superblocks in A have eigenvalue 0 and therefore $\lambda(A)$ is the only eigenvalue of A.

It also implies that all nodes in A are critical and by Corollary 3.6.1 and Proposition 3.7.1, we have $V(A) = Im(\Gamma(A))$. And using Theorem 3.4.5, it immediately implies that

$$V^{+}(A) = \{\Gamma(A) \otimes x \mid x \in \mathbb{R}^n\}.$$

$\square$

Using Proposition 6.4.2, 6.4.3 and 6.4.4, we can generate a subroutine which converts any doubly $\mathbb{R}$-astic matrix into a strongly definite matrix and the original problem is not affected.

First the subroutine will need to find an optimal permutation for A, namely π and this can be found by using the Hungarian method.

The next step will be to permute the columns of the matrix so that the identity becomes an optimal permutation. This can be done by using the permutation π which we obtained earlier and apply this permutation to the matrix A and obtain a new matrix A', i.e. we let $\forall j \in N$, $a'_{i,\pi(j)} := a_{ij}$, $\forall i \in M$.

Finally we will subtract from every column of the matrix by a constant such that the diagonal entry becomes zero, i.e. we let $\forall j \in N$, $a'_{ij} := a'_{ij} - a'_{jj}$, $\forall i \in M$. Then we can see that $\forall i \in N$, $a'_{ii} = 0$ and $id \in ap(A')$ and hence A' is strongly definite.

Note that it is possible that we can permute the columns and subtract from each column a constant at the same time and for simplicity, this will be done in the following subroutine.

Subroutine 1.

Input: A doubly $\mathbb{R}$-astic matrix $A \in \overline{\mathbb{R}}^{m \times n}$.

Output: A strongly definite matrix $A' \in \overline{\mathbb{R}}^{m \times n}$.

Let π be an optimal permutation of A found by the Hungarian method.

for $j = 1$ to n do

begin

for all $i \in M$, set $a'_{i,\pi(j)} := a_{ij} - a_{\pi(j),j}$.

end

Therefore without loss of generality, we can consider strongly definite matrices only.

6.4.2 Integer Simple Image Set

We will first investigate the integer simple image set of a strongly regular matrix. We will start by defining the integer simple image set.

Definition 6.4.3. If $A \in \overline{\mathbb{R}}^{n \times n}$ is strongly regular, then $IS_A = \{b \in \mathbb{Z}^m \mid b \in S_A\}$, i.e. the set of integer vectors in the simple image set of A.

The next step will be to find conditions which a strongly regular matrix has to satisfy such that an integer image exists.

Proposition 6.4.6. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly regular, strongly definite and $C = (\lceil a_{ij} \rceil)$. If $IS_A \neq \emptyset$ then $\forall \sigma \in P_n$,

$$\sum_{j\in N} c_{j,\sigma(j)} \leq -|I_\sigma| \tag{6.13}$$

where $I_\sigma = \{j = \sigma(i) \in N \mid \sigma(i) \neq i \text{ and } a_{i,\sigma(i)} \in \mathbb{Z}\}$.

Proof. Let $b = (b_1, ..., b_n) \in IS_A$, then we know

$$\begin{aligned} a_{ij} - b_i &< a_{jj} - b_j, \quad \forall i \neq j \\ a_{ij} - b_i + b_j &< 0, \qquad\qquad \forall i \neq j. \end{aligned} \tag{6.14}$$

If $a_{ij} \in \mathbb{Z}$, then

$$a_{ij} - b_i + b_j \leq -1, \quad \forall a_{ij} \in \mathbb{Z}, i \neq j$$

since $b \in \mathbb{Z}^n$. Hence (6.14) becomes

$$\begin{aligned} c_{ij} - b_i + b_j &< \quad 0, \quad \forall a_{ij} \notin \mathbb{Z} \\ c_{ij} - b_i + b_j &\leq -1, \quad \forall a_{ij} \in \mathbb{Z}, i \neq j \end{aligned} \tag{6.15}$$

Let σ be any permutation, then we have $\forall i \in N$, $\exists j \in N$ such that $\sigma(i) = j$. Now summing over all i in (6.15), we have

$$\sum_{i\in N} c_{i,\sigma(i)} \leq -|I_\sigma|.$$

$\square$

The above proposition gives us a necessary condition for the existence of an integer image in the case of a strongly regular matrix. Unfortunately it is not practical to check for every permutation and find out if this condition is satisfied. Therefore we will modify the above condition with the help of the following definition.

Definition 6.4.4. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly definite. Then define $\widetilde{C}$ to be the following matrix:

$$\widetilde{C} = (\widetilde{c}_{ij}) = \begin{cases} \lceil a_{ij} \rceil + 1 & \text{if } i \in I_j(A, j), i \neq j \\ \lceil a_{ij} \rceil & \text{otherwise.} \end{cases}$$

This is the same as taking the upper integer value for all entries in A but we will increase the values of all the off-diagonal entries by one if they are already integer.

We can see that the matrix $\widetilde{C}$ is similar to the C matrix which was defined in Proposition 6.4.6.

Corollary 6.4.3. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly regular and strongly definite. If $IS_A \neq \emptyset$ then $\widetilde{C}$ is definite.

Proof. By Proposition 6.4.6, we know that if $IS_A \neq \emptyset$ then $\forall \sigma \in P_n$,

$$\sum_{i\in N} c_{i,\sigma(i)} \leq -|I_\sigma|,$$

where $C = (\lceil a_{ij} \rceil)$. Since $\forall \sigma \in P_n$

$$\sum_{i\in N} c_{i,\sigma(i)} + |I_\sigma| \leq 0,$$

then we can deduce that

$$\sum_{i\in N} \widetilde{c}_{i,\sigma(i)} \leq 0.$$

We also know that $\widetilde{c}_{ii} = c_{ii} = 0, \forall i$, hence $\widetilde{C}$ is a definite matrix. $\square$

Corollary 6.4.3 gives us another necessary condition which can be tested more efficiently. The next question will be to find out if this necessary condition is also sufficient. To answer this question, we define the integer eigenspace of a matrix.

Definition 6.4.5. Let $A \in \overline{\mathbb{R}}^{n\times n}$, then $IV(A) = \{b \in \mathbb{Z}^m \mid b \in V(A)\}$, i.e. the set of integer eigenvectors of A.

For simplicity, we will also define the set $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{\epsilon\}$. Now we will look at the following proposition.

Proposition 6.4.7. Let $A \in \overline{\mathbb{Z}}^{m\times n}$ be doubly $\mathbb{R}$-astic, then $IIm(A) \neq \emptyset$ and furthermore, $IIm(A) = \{A \otimes x \mid x \in \mathbb{Z}^n\}$.

Proof. Let x be the zero vector, i.e. $x = (0, ..., 0)^T$, then $\forall i \in M$,

$$b_i = \max_{j=1,\ldots,n} (a_{ij} + 0) > \epsilon$$

since A is doubly $\mathbb{R}$-astic. Also $A \in \overline{\mathbb{Z}}^{m\times n}$ and therefore $\forall i \in M$, $b_i \in \mathbb{Z}$. Hence $IIm(A) \neq \emptyset$.

Let $x \in \mathbb{Z}^n$ and $b = A \otimes x$. Then it immediately follows that $b \in IIm(A)$ and hence $\{A \otimes x \mid x \in \mathbb{Z}^n\} \subseteq IIm(A)$.
Let $b \in IIm(A)$ and $\bar{x} = A^* \otimes' b$, then $\forall j \in N$

$$\bar{x}_j = \min_{i=1,\ldots,m} (a^*_{ji} + b_i)$$

$$\bar{x}_j = \min_{i=1,\ldots,m} (-a_{ij} + b_i)$$

Since $a_{ij}, b_i \in \overline{\mathbb{Z}}$, $\forall i \in M$ and A is doubly $\mathbb{R}$-astic, it implies that $\bar{x}_j \in \mathbb{Z}$, $\forall j \in N$. Using Corollary 2.3.3, we have that $A \otimes \bar{x} = b$ and therefore $IIm(A) \subseteq \{A \otimes x \mid x \in \mathbb{Z}^n\}$. Hence $IIm(A) = \{A \otimes x \mid x \in \mathbb{Z}^n\}$. □

Proposition 6.4.8. Let $A \in \mathbb{Z}^{n\times n}$, then

$$IV(A) \neq \emptyset \iff \lambda(A) \in \mathbb{Z}.$$

Proof. Since $A \in \mathbb{Z}^{n\times n}$, then it immediately implies that A is irreducible and $\Lambda(A) = \{\lambda(A)\}$. If $IV(A) \neq \emptyset$, then $\exists x \in \mathbb{Z}^n$ such that $A \otimes x = \lambda(A) \otimes x$. Since both A and x are integer, it immediately implies that $\lambda(A) \in \mathbb{Z}$.

Suppose that $\lambda(A) \in \mathbb{Z}$, then $\Gamma(\lambda(A)^{-1} \otimes A) \in \mathbb{Z}^{n\times n}$ and therefore all fundamental eigenvectors are integer. Hence $IV(A) \neq \emptyset$. $\square$

Corollary 6.4.4. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly definite. Suppose that $\Gamma(A) \in \overline{\mathbb{Z}}^{n\times n}$, then $IV(A) \neq \emptyset$ and furthermore, $IV(A) = \{\Gamma(A) \otimes x \mid x \in \mathbb{Z}^n\}$.

Proof. It immediately follows from Proposition 6.4.5 and Proposition 6.4.7. $\square$

Proposition 6.4.9. Suppose that $A \in \overline{\mathbb{R}}^{n\times n}$ is strongly definite, then $\Gamma(A)$ is doubly $\mathbb{R}$-astic.

Proof. By the definition of strongly definite matrices, we know that $\forall i \in N$, $a_{ii} = 0$ and $\lambda(A) = 0$. Therefore by the definition of metric matrices, we know that

$$\Gamma(A) = A \oplus A^2 \oplus ... \oplus A^n$$

and hence $\forall i \in N$,

$$(\Gamma(A))_{ii} = a_{ii} \oplus a_{ii}^2 \oplus ... \oplus a_{ii}^n = 0.$$

It implies that $\Gamma(A)$ contains no ϵ rows or columns and therefore it is doubly $\mathbb{R}$-astic. $\square$

Proposition 6.4.10. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly definite. If $\widetilde{C}$ is definite, then $\widetilde{C}$ is also strongly definite. Similarly if C is definite, then C is also strongly definite.

Proof. It immediately follows from the definition of $\widetilde{C}$, C and strongly definite matrices. $\square$

Using the above results, we will obtain the following two important corollaries.

Corollary 6.4.5. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly definite, if $\widetilde{C}$ is definite, then $IV(\widetilde{C}) \neq \emptyset$, i.e. it has integer eigenvectors.

Proof. By Proposition 6.4.10, we know that $\widetilde{C}$ is strongly definite. Then by Proposition 6.4.9, we also know that $\Gamma(\widetilde{C})$ is doubly $\mathbb{R}$-astic and it immediately follows from Corollary 6.4.4 that $IV(\widetilde{C}) \neq \emptyset$. $\square$

Corollary 6.4.6. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly definite. If C is definite, then $IV(C) \neq \emptyset$, i.e. it has integer eigenvectors.

Proof. It immediately follows Corollary 6.4.4, Proposition 6.4.9 and 6.4.10. $\square$

Using the results above, we can obtain and prove the following proposition.

Proposition 6.4.11. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly regular and strongly definite. If $\widetilde{C}$ is definite then $IV(\widetilde{C}) \subseteq IS_A$ and hence $IS_A \neq \emptyset$.

Proof. By Corollary 6.4.5, we know that $IV(\widetilde{C}) \neq \emptyset$. Let $b \in IV(\widetilde{C})$, then we have

$$\begin{gathered}\widetilde{C} \otimes b = b \\ \max_{j=1,\ldots,n} (\widetilde{c}_{ij} + b_j) = b_i, \quad \forall i \in N \\ \widetilde{c}_{ij} + b_j \leq b_i, \quad \forall i,j \in N\end{gathered} \tag{6.16}$$

If $i \notin I_j(A,j)$, then $f(a_{ij}) \neq f(a_{jj})$ and hence

$$\widetilde{c}_{ij} = \lceil a_{ij} \rceil > a_{ij}, \quad \forall i \notin I_j(A,j). \tag{6.17}$$

Suppose that $i \in I_j(A,j) - \{j\}$, then

$$\widetilde{c}_{ij} = \lceil a_{ij} \rceil + 1 > \lceil a_{ij} \rceil = a_{ij}. \tag{6.18}$$

By (6.17) and (6.18), we will have

$$\widetilde{c}_{ij} > a_{ij}, \quad \forall i \neq j. \tag{6.19}$$

Using (6.16) and (6.19), we have $\forall j \in N$,

$$a_{ij} + b_j < b_i, \quad \forall i \neq j$$
$$a_{ij} - b_i < -b_j, \quad \forall i \neq j.$$

This implies that $b \in IS_A$ and hence $IV(\widetilde{C}) \subseteq IS_A$. Since $IV(\widetilde{C}) \neq \emptyset$, then $IS_A \neq \emptyset$. $\square$

From Proposition 6.4.11, we have found that $IV(\widetilde{C})$ is a subset of IS_A. And we have also shown that IS_A is not empty, therefore the definiteness of $\widetilde{C}$ is a sufficient condition for the existence of an integer image. Next we will show that $IS_A = IV(\widetilde{C})$.

Proposition 6.4.12. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly regular and strongly definite. If $\widetilde{C}$ is definite then $IS_A \subseteq IV(\widetilde{C})$.

Proof. Suppose that $b \in IS_A$, then by (6.15) we have

$$\begin{aligned} c_{ij} - b_i + b_j &\leq 0, \quad \forall a_{ij} \notin \mathbb{Z} \\ c_{ij} - b_i + b_j &\leq -1, \quad \forall a_{ij} \in \mathbb{Z}, i \neq j \end{aligned}$$

where $c_{ij} = \lceil a_{ij} \rceil, \forall i, j$. Hence

$$\begin{aligned} c_{ij} + b_j &\leq b_i, \quad \forall a_{ij} \notin \mathbb{Z} \\ c_{ij} + 1 + b_j &\leq b_i, \quad \forall a_{ij} \in \mathbb{Z}, i \neq j. \end{aligned}$$

These imply

$$\widetilde{c}_{ij} + b_j \leq b_i \;\; \forall i, j \in N$$
$$\max_{j=1,\ldots,n} (\widetilde{c}_{ij} + b_j) = b_i \;\; \forall i \in N$$
$$\widetilde{C} \otimes b = b.$$

Since $\widetilde{C}$ is definite, this implies that $b \in IV(\widetilde{C})$. Therefore $IS_A \subseteq IV(\widetilde{C})$. $\square$

Corollary 6.4.7. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly regular and strongly definite. Then the following are equivalent:

1. $IV(\widetilde{C}) = IS_A$ and $IS_A \neq \emptyset$.

2. $\widetilde{C}$ is definite.

Proof. It immediately follows from Corollary 6.4.3, Proposition 6.4.11 and 6.4.12. $\square$

Example 6.4.3. Let

$$A = \begin{pmatrix} 0 & 0.4 & -1.5 & -2 \\ -1.5 & 0 & -2 & -3 \\ -1.5 & -1 & 0 & -2.5 \\ -2 & -1 & -2.1 & 0 \end{pmatrix},$$

then identity is the unique optimal permutation for A and hence A is strongly regular. Now we have

$$\widetilde{C} = \begin{pmatrix} 0 & 1 & -1 & -1 \\ -1 & 0 & -1 & -2 \\ -1 & 0 & 0 & -2 \\ -1 & 0 & -2 & 0 \end{pmatrix}.$$

We can see that $\widetilde{C}$ contains only one positive entry, namely $\widetilde{c}_{12}$, and all cycles containing this entry have non-positive weight. There are no positive cycles and all diagonal entries are zeros. Therefore we can deduce that $\widetilde{C}$ is strongly definite. Using this, we can obtain the metric matrix of $\widetilde{C}$ which is

$$\widetilde{C} \oplus \widetilde{C}^2 \oplus \widetilde{C}^3 = \begin{pmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & -1 & -2 \\ -1 & 0 & 0 & -2 \\ -1 & 0 & -1 & 0 \end{pmatrix},$$

We can see that all four columns of the metric matrix are the fundamental eigenvectors of $\widetilde{C}$ since the diagonal entries of each column are equal to zero. We can also see that the first two columns are equivalent as they are a multiple of each other. Then the non-equivalent fundamental eigenvectors of $\widetilde{C}$ are

$$\begin{pmatrix} 0 \\ -1 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ 0 \\ -1 \end{pmatrix} \text{ and } \begin{pmatrix} -1 \\ -2 \\ -2 \\ 0 \end{pmatrix}.$$

If we take any linear combination of the three eigenvectors, then $b \in IS_A$. For example, let

$$b = (0 \otimes \begin{pmatrix} 0 \\ -1 \\ -1 \\ -1 \end{pmatrix}) \oplus (1 \otimes \begin{pmatrix} 0 \\ -1 \\ 0 \\ -1 \end{pmatrix}) \oplus (2 \otimes \begin{pmatrix} -1 \\ -2 \\ -2 \\ 0 \end{pmatrix}) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 2 \end{pmatrix},$$

hence set $x = (1, 0, 1, 2)^T$. Then $b \in IS_A$. This fact can be checked by considering the following matrix

$$a_{ij} - b_i = \begin{pmatrix} 0-1 & 0.4-1 & -1.5-1 & -2-1 \\ -1.5-0 & 0-0 & -2-0 & -3-0 \\ -1.5-1 & -1-1 & 0-1 & -2.5-1 \\ -2-2 & -1-2 & -2.1-2 & 0-2 \end{pmatrix}$$
$$= \begin{pmatrix} -1 & -0.6 & -2.5 & -3 \\ -1.5 & 0 & -2 & -3 \\ -2.5 & -2 & -1 & -3.5 \\ -4 & -3 & -4.1 & -2 \end{pmatrix}$$

We can immediately see that the diagonal entries are strictly greater than all other entries in the same column. Therefore b is an element of the integer simple image set.

Corollary 6.4.7 gives us a necessary and sufficient condition for an integer simple image set to be non-empty for strongly regular and strongly definite matrices. It also tells us that the integer simple image set is exactly the set of integer eigenvectors of the matrix $\widetilde{C}$ which arises from A. Since the simple image set of any strongly regular matrices is a subset of their integer image set, the next natural step will be to investigate the relationship between the integer simple image set and integer image set.

6.4.3 Integer Image Set

Now we will investigate the integer image set of a strongly regular matrix. The simple image set of a matrix is a subset of its image set. Therefore by using previous results, we can immediately obtain a sufficient condition for integer image set to be non-empty.

Corollary 6.4.8. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly regular and strongly definite. If $\widetilde{C}$ is definite then $IV(\widetilde{C}) \subseteq IIm(A) \neq \emptyset$.

Proof. It immediately follows from Proposition 6.4.11. □

The result above can be generalized by relaxing some of the initial assumptions.

Proposition 6.4.13. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly definite and $C = (\lceil a_{ij} \rceil)$. If C is definite, then $IV(C) \subseteq IIm(A) \neq \emptyset$.

Proof. Since C is a definite and integer matrix, by Corollary 6.4.6 we know that $IV(C) \neq \emptyset$. Let $b \in IV(C)$, then we have

$$\begin{aligned} C \otimes b &= b \\ \max_{j=1,\ldots,n} (c_{ij} + b_j) &= b_i, \quad \forall i \in N \qquad (6.20) \\ c_{ij} + b_j &\leq b_i, \quad \forall i, j \in N. \end{aligned}$$

Since

$$c_{ij} \geq a_{ij}, \quad \forall i, j, \tag{6.21}$$

using (6.20) and (6.21), we have $\forall j \in N$,

$$\begin{aligned} a_{ij} + b_j &\leq b_i, \quad \forall i \\ a_{ij} - b_i &\leq a_{jj} - b_j, \quad \forall i \end{aligned} \tag{6.22}$$

since $a_{jj} = 0$ for all j. This implies that $j \in M_j(A, b)$, $\forall j$ and hence

$$\bigcup_{j \in N} M_j(A, b) = N.$$

Therefore $b \in IIm(A)$ and hence $IV(C) \subseteq IIm(A) \neq \emptyset$. $\square$

The above proposition gives us a sufficient condition for any matrix to have an integer image. The next step is to decide if this sufficient condition is also a necessary condition for the existence of an integer image. Unfortunately this is not the case in general; this can be seen by the following two examples.

Example 6.4.4. Let

$$A = \begin{pmatrix} 0 & -2 & -1 \\ 0 & 0 & 0.1 \\ -1.1 & -0.1 & 0 \end{pmatrix},$$

then identity is an optimal permutation for A. Note that $a_{23} + a_{32} = 0$ and therefore identity is not the unique permutation and hence A is not strongly regular. Now we have

$$C = \begin{pmatrix} 0 & -2 & -1 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix}$$

which is not definite.

Let $x = (0, -0.9, -1)^T$, then $A \otimes x$ will be

$$\begin{pmatrix} 0 & -2 & -1 \\ 0 & 0 & 0.1 \\ -1.1 & -0.1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ -0.9 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

and thus shows that a strongly definite matrix may have an integer image even if C is not definite.

Example 6.4.5. Let

$$A = \begin{pmatrix} 0 & 1.4 & -1.5 & -2 \\ -1.5 & 0 & -2 & -3 \\ -1.5 & -1 & 0 & -2.5 \\ -2 & -1 & -2.1 & 0 \end{pmatrix},$$

then identity is the unique optimal permutation for A and hence A is strongly regular and strongly definite. Now we have

$$\widetilde{C} = \begin{pmatrix} 0 & 2 & -1 & -1 \\ -1 & 0 & -1 & -2 \\ -1 & 0 & 0 & -2 \\ -1 & 0 & -2 & 0 \end{pmatrix}, C = \begin{pmatrix} 0 & 2 & -1 & -2 \\ -1 & 0 & -2 & -3 \\ -1 & -1 & 0 & -2 \\ -2 & -1 & -2 & 0 \end{pmatrix}$$

and $\lambda(\widetilde{C}) = \lambda(C) = \frac{1}{2}(a_{12} + a_{21}) = \frac{1}{2}$. Therefore both $\widetilde{C}$ and C are not definite.

Let $x = (0, -1.4, -1, 0)^T$, then $A \otimes x$ will be

$$\begin{pmatrix} 0 & 1.4 & -1.5 & -2 \\ -1.5 & 0 & -2 & -3 \\ -1.5 & -1 & 0 & -2.5 \\ -2 & -1 & -2.1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ -1.4 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix}$$

and therefore $IIm(A) \neq \emptyset$. This shows that a strongly regular matrix may have an integer image even if $\widetilde{C}$ or C is not definite.

From the two examples above, we know that the definitiveness of the matrix C is not a necessary condition for integer image set to be non-empty for a matrix in general.

Now let us look at Proposition 6.4.1 again, we have seen that there is a close relationship between the integer simple image set and integer image set of any typical matrices. Now we would like to find out if there exists any other special cases of strongly regular matrices which share the same property as typical matrices.

Proposition 6.4.14. Let $A \in \overline{\mathbb{R}}^{n \times n}$ be strongly regular and strongly definite. If $\forall j \in N$, $|I_j(A, j)| = 1$, i.e. all off-diagonal entries are non-integer, then $IIm(A) = IS_A$.

Proof. Let $b \in IIm(A)$ and suppose that $b \notin IS_A$. Then $\exists i_1, i_2 \in N$, $i_1 \neq i_2$ such that $i_1 \in M_{i_2}(A, b)$, i.e.

$$a_{i_1 i_2} - b_{i_1} \geq a_{i_2 i_2} - b_{i_2} = -b_{i_2}.$$

Since $b \in \mathbb{Z}^n$ and $f(a_{i_1 i_2}) \neq f(a_{i_2 i_2}) = 0$, this implies that both sides cannot be equal. Therefore we have

$$a_{i_1 i_2} - b_{i_1} > -b_{i_2}.$$

Since $b \in Im(A)$, this implies that $\exists i_3 \in N$, $i_3 \neq i_2$ such that $i_2 \in M_{i_3}(A, b)$, i.e.

$$a_{i_2 i_3} - b_{i_2} \geq a_{i_3 i_3} - b_{i_3} = -b_{i_3}.$$

By using the same argument as before, we will then have

$$a_{i_2 i_3} - b_{i_2} > -b_{i_3}.$$

We will now repeat this process until we obtain

$$a_{i_k i_{k+1}} - b_{i_k} > -b_{i_{k+1}}$$

where i_{k+1} is one of the previous indexes and without loss of generality, we can assume $i_{k+1} = i_1$. Now if we sum up all the inequalities, we will then obtain

$$a_{i_1 i_2} + a_{i_2 i_3} + ... + a_{i_k i_1} > 0.$$

Since the weight of the identity permutation is equal to zero, this implies that there exists a heavier permutation than the identity permutation and hence $id \notin ap(A)$. This is a contradiction and therefore $b \in IS_A$ and hence $IIm(A) \subseteq IS_A$. Since $S_A \subseteq Im(A)$, it immediately follows that $IS_A \subseteq IIm(A)$ and hence $IIm(A) = IS_A$. $\square$

Note that the matrix C is closely related to the matrix $\widetilde{C}$. In fact the two matrices are equal if the fractional part of the diagonal entries are different than any other entries on the same column, i.e. $\forall j \in N$, $|I_j(A, j)| = 1$. Using this fact, we can then combine the above proposition with Corollary 6.4.7 and obtain the following corollary.

Corollary 6.4.9. Let $A \in \overline{\mathbb{R}}^{n \times n}$ be strongly regular, strongly definite and $C = (\lceil a_{ij} \rceil)$. Suppose that $\forall j \in N$, $|I_j(A, j)| = 1$, then $IV(C) = IIm(A)$. Furthermore, C is definite if and only if $IIm(A) \neq \emptyset$.

Proof. It immediately follows from Corollary 6.4.7 and Proposition 6.4.14. $\square$

The above corollary gives us a special case when we can obtain the set of integer images.

By Corollary 6.4.4, we can describe the set of integer images explicitly.

Corollary 6.4.10. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly regular, strongly definite and $C = (\lceil a_{ij} \rceil)$. Suppose that $\forall j \in N$, $|I_j(A,j)| = 1$, then C is definite if and only if

$$IIm(A) = \{\Gamma(C) \otimes x \mid x \in \mathbb{Z}^n\}.$$

Proof. It immediately follows from Corollary 6.4.9 and Corollary 6.4.4. $\square$

Previously we have seen that Proposition 6.4.13 gives us a sufficient condition for a strongly definite matrix to have integer images. Next we would like to find out if the matrix still have integer images when this condition is not satisfied.

Definition 6.4.6. Let $A \in \overline{\mathbb{R}}^{n\times n}$, $R = \{r_1, r_2, ..., r_\mu\} \subseteq N$ where $1 \leq r_1 < r_2 < ... < r_\mu \leq n$. Then we will denote $A[R]$ to be a principal sub-matrix of A where

$$A[R] = \begin{pmatrix} a_{r_1r_1} & a_{r_1r_2} & \dots & a_{r_1r_\mu} \\ a_{r_2r_1} & a_{r_2r_2} & \dots & a_{r_2r_\mu} \\ \dots & \dots & \dots & \dots \\ a_{r_\mu r_1} & a_{r_\mu r_2} & \dots & a_{r_\mu r_\mu} \end{pmatrix}.$$

Proposition 6.4.15. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly definite, then $IIm(A) \neq \emptyset$ if $\exists R \subseteq N$, $R \neq \emptyset$ such that

1) $C[R]$ is definite,

2) $\forall s \notin R, \exists r(s) \in R$ such that $a_{sr(s)} \in \mathbb{Z}$ and $a_{r(s)j} \geq a_{sj} - a_{sr(s)} \forall j \in R$.

Proof. Suppose that $\exists R \subseteq N$ such that 1) and 2) hold. By Corollary 6.4.6 we know that

$IV(C[R]) \neq \emptyset$. Let $b \in \mathbb{R}^n$ be the vector such that $b[R] \in IV(C[R])$ and $\forall s \notin R$,

$$b_s = a_{sr(s)} + b_{r(s)} \tag{6.23}$$

where s and $r(s)$ satisfy $2)$.

Since $b[R] \in IV(C[R])$ and $C[R]$ is definite, by Proposition 6.4.13 we know that $b[R] \in IIm(A[R])$. Furthermore, by (6.22) we also know that $\forall j \in R$

$$a_{ij} - b_i \leq -b_j, \quad \forall i \in R. \tag{6.24}$$

Now let $s \notin R$, then by $2)$ we know that $\exists r(s) \in R$ such that

$$\begin{aligned} a_{r(s)j} &\geq a_{sj} - a_{sr(s)}, \\ a_{r(s)j} &\geq a_{sj} - (b_s - b_{r(s)}), \text{ by } (6.23) \\ a_{r(s)j} - b_{r(s)} &\geq a_{sj} - b_s. \end{aligned} \tag{6.25}$$

Since $r(s) \in R$, then using (6.24), we know that $\forall j \in R$,

$$-b_j \geq a_{r(s)j} - b_{r(s)}$$

and hence $\forall j \in R$,

$$-b_j \geq a_{sj} - b_s. \tag{6.26}$$

Therefore (6.24) and (6.26) imply that $j \in M_j(A, b)$, $\forall j \in R$. By (6.23), we also know that

$\forall s \notin R$,

$$b_s = a_{sr(s)} + b_{r(s)},$$
$$-b_{r(s)} = a_{sr(s)} - b_s. \tag{6.27}$$

This implies that $\forall s \notin R$, $\exists r(s) \in R$ such that $s \in M_{r(s)}(A, b)$. Therefore

$$\bigcup_{j \in R} M_j(A, b) = M$$

and hence $b \in Im(A)$. We have $\forall j \in R$, $b_j \in \mathbb{Z}$ and also $\forall s \notin R$, $b_s \in \mathbb{Z}$. Since $a_{sr(s)}$ and $b_{r(s)}$ are integers, we then have $b \in \mathbb{Z}^n$ and hence $b \in IIm(A)$. $\square$

The above proposition gives us another sufficient condition for the existence of an integer image in a strongly definite matrix. Using a similar idea and the following definitions, we can present another sufficient condition.

Definition 6.4.7. Let $A \in \overline{\mathbb{R}}^{n \times n}$, $r_l \in R = \{r_1, r_2, ..., r_\mu\} \subseteq N$ and $s \notin R$, then we will denote $A[R, r_l, s]$ to be the matrix

$$(A[R, r_l, s])_{ij} = \begin{cases} a_{sr_j} - a_{sr_l} & \text{if } i = l, \\ a_{r_i r_j} & \text{otherwise.} \end{cases}$$

Hence

$$A[R, r_l, s] = \begin{pmatrix} a_{r_1 r_1} & a_{r_1 r_2} & ... & a_{r_1 r_\mu} \\ ... & ... & ... & ... \\ a_{sr_1} - a_{sr_l} & a_{sr_2} - a_{sr_l} & ... & a_{sr_\mu} - a_{sr_l} \\ ... & ... & ... & ... \\ a_{r_\mu r_1} & a_{r_\mu r_2} & ... & a_{r_\mu r_\mu} \end{pmatrix}.$$

Furthermore, we will denote $C[R, r_l, s]$ to be the matrix devised from $A[R, r_l, s]$ by taking the upper integer part of all entries in $A[R, r_l, s]$.

Proposition 6.4.16. Let $A \in \overline{\mathbb{R}}^{n\times n}$ be strongly definite, then $IIm(A) \neq \emptyset$ if $\exists R \subseteq N$, $R \neq \emptyset$ and $S = \{s_1, ..., s_\nu\} = N - R$ such that

1) $C[R]$ is definite,
2) $\forall s \in S, \exists r(s) \in R$ such that $a_{sr(s)} \in \mathbb{Z}$,
3) $\forall s_i \in S, IV(C[R, r(s_i), s_i])$ is definite, and
4) $IV(C[R, r(s_1), s_1]) \cap ... \cap IV(C[R, r(s_\nu), s_\nu]) \cap IV(C[R]) \neq \emptyset$.

Proof. Let U be the intersection in $4)$ and suppose that $\exists R \subseteq N$ such that $1)$, $2)$, $3)$ and $4)$ hold. Then we let $b \in \mathbb{R}^n$ be the vector such that $b[R] \in U$ and $\forall s \in S$,

$$b_s = a_{sr(s)} + b_{r(s)} \tag{6.28}$$

where s and $r(s)$ satisfy $2)$, $3)$ and $4)$.
Since $b[R] \in IV(C[R])$ and $C[R]$ is definite, by Proposition 6.4.13 we know that $b[R] \in IIm(A[R])$. Furthermore, by (6.22) we also know that

$$a_{r_i r_j} - b_{r_i} \leq -b_{r_j}, \quad \forall r_i, r_j \in R. \tag{6.29}$$

We also know that $b[R] \in IV(C[R, r(s), s])$, then we have

$$C[R, r(s), s] \otimes b[R] = b[R].$$

Now if we consider the row of the above linear system which corresponds to the $r(s)^{th}$ row of C, we will obtain

$$\begin{aligned}
\max_{r_j \in R}((c_{sr_j} - c_{sr(s)}) + b_{r_j}) &= b_{r(s)}, \\
\max_{r_j \in R}(\lceil(a_{sr_j}\rceil - \lceil a_{sr(s)})\rceil + b_{r_j}) &= b_{r(s)}, \\
\lceil(a_{sr_j} - a_{sr(s)})\rceil + b_{r_j} &\leq b_{r(s)},\ \forall r_j \in R \text{ since } a_{sr(s)} \in \mathbb{Z}.
\end{aligned} \tag{6.30}$$

Since $a_{sr(s)} \in \mathbb{Z}$ and $\lceil a_{sr_j}\rceil \geq a_{sr_j}$, then we have

$$\lceil(a_{sr_j} - a_{sr(s)})\rceil \geq a_{sr_j} - a_{sr(s)}. \tag{6.31}$$

Using (6.30) and (6.31), we will have obtained $\forall r_j \in R$,

$$\begin{aligned}
a_{sr_j} - a_{sr(s)} + b_{r_j} &\leq b_{r(s)}, \\
a_{sr_j} - a_{sr(s)} - b_{r(s)} &\leq -b_{r_j}, \\
a_{sr_j} - b_s &\leq -b_{r_j},\ \text{by (6.28)}.
\end{aligned} \tag{6.32}$$

Therefore (6.29) and (6.32) implies that $r_j \in M_{r_j}(A, b)$, $\forall r_j \in R$. By (6.28), we also know that $\forall s \in S$

$$\begin{aligned}
b_s &= a_{sr(s)} + b_{r(s)}, \\
-b_{r(s)} &= a_{sr(s)} - b_s.
\end{aligned} \tag{6.33}$$

This implies that $\forall s \in S$, $\exists r(s) \in R$ such that $s \in M_{r(s)}(A, b)$. Therefore

$$\bigcup_{j \in R} M_j(A, b) = M$$

and hence $b \in Im(A)$. We know that $\forall r_j \in R$, $b_{r_j} \in \mathbb{Z}$ and we have $\forall s \in S$, $b_s \in \mathbb{Z}$. Since $a_{sr(s)}$ and $b_{r(s)}$ are integers, then $b \in \mathbb{Z}^n$ and hence $b \in IIm(A)$. $\square$

6.5 The General Case

In this section, we will consider the task of deciding whether the integer image of a matrix exists for the general case. Let us first consider the following proposition.

Proposition 6.5.1. Let $A \in \mathbb{R}^{m\times n}$, then $\forall b \in IIm(A)$,

$$\lceil \min_{k=1,\dots,n}(a_{ik} - a_{jk})\rceil \leq b_i - b_j \leq \lfloor \max_{k=1,\dots,n}(a_{ik} - a_{jk})\rfloor, \ \ \forall i, j \in M.$$

Proof. Let $b \in IIm(A)$, then by Corollary 2.3.1, $\forall i \in M$, $\exists r \in N$ such that $i \in M_r(A, b)$. Hence $\forall j \in M$,

$$\begin{aligned} a_{ir} - b_i &\geq a_{jr} - b_j, \\ a_{ir} - a_{jr} &\geq b_i - b_j, \\ \max_{k=1,\dots,n}(a_{ik} - a_{jk}) &\geq a_{ir} - a_{jr}, \\ &\geq b_i - b_j. \end{aligned}$$

But since $b \in \mathbb{Z}^m$, we have $\forall i, j \in M$,

$$\lfloor \max_{k=1,\dots,n}(a_{ik} - a_{jk})\rfloor \geq b_i - b_j.$$

Similarly we also have $\forall i, j \in M$,

$$\begin{aligned} \max_{k=1,\dots,n}(a_{jk} - a_{ik}) &\geq b_j - b_i, \\ -\max_{k=1,\dots,n}(a_{jk} - a_{ik}) &\leq b_i - b_j, \\ \min_{k=1,\dots,n}(a_{ik} - a_{jk}) &\leq b_i - b_j, \end{aligned}$$

and hence $\forall i, j \in M$,

$$\lceil \min_{k=1,\dots,n} (a_{ik} - a_{jk}) \rceil \leq b_i - b_j.$$

$\square$

Using the above proposition, we can obtain a bound between any two components for any integer image of a real matrix. This provides a necessary condition for the existence of an integer image for a matrix. Therefore an integer image of a matrix can only exist when for each bound, there exists at least an integer value satisfying the bounds.

Since the matrix is real and the difference between any two components of an integer image must be integer, the number of possibilities for each bound is finite. We can also assume without loss of generality, that one component of b, say b_1, is zero. Using these facts, we can generate an algorithm to check for the existence of integer images for any real matrices.

Algorithm 4.

Input: A matrix $A \in \mathbb{R}^{m \times n}$.

Output: A vector $b \in \mathbb{Z}^n$ for which $b \in IIm(A)$ or an indication that such a b does not exist.

Set $l_1 := 0$ and $u_1 := 0$.

for $i = 2$ to m do

begin

Set

$$l_i := \lceil \min_{k=1,\dots,n} (a_{ik} - a_{1k}) \rceil$$

$$u_i := \lfloor \max_{k=1,\dots,n} (a_{ik} - a_{1k}) \rfloor.$$

Set $d_i := u_i - l_i + 1$.

if $d_i \leq 0$ then no integer image exists. Stop.

end

Set $b^{(1)} := l$ and

$$D := \prod_{i=2}^{m} d_i.$$

for $i = 1$ to D do

begin

if $b^{(i)} = A \otimes (A^* \otimes' b^{(i)})$

Integer image exists and $b := b^{(i)}$ is an integer image of A. Stop.

else

Set $b^{(i+1)} := b^{(i)}$ and $b_m^{(i+1)} := b_m^{(i)} + 1$.

for $j = m$ to 2 do

begin

if $b_j^{(i+1)} > u_j$

Set $b_j^{(i+1)} := l_j$ and $b_{j-1}^{(i+1)} := b_{j-1}^{(i+1)} + 1$.

end

end

The above algorithm generates all possible candidates which can be an integer image and checks if any of the integer vectors are indeed an integer image of A. We can see that the main loop is repeated at most D times and D is determined by calculating the difference between the entries in the first row with corresponding entries in other row. We can reduce the number of times the loop is repeated by finding a row which generates a smaller D. This can be done by the following subroutine.

Subroutine 2.

Input: A matrix $A \in \mathbb{R}^{m \times n}$.

Output: A row index p, $D \in \mathbb{Z}$, $l, u \in \mathbb{Z}^m$ and $A' \in \mathbb{R}^{m\times n}$.

for $i = 1$ to m do

begin

for $j = 1$ to m, $j \neq i$ do

begin

Set

$$l_{ij} := \lceil \min_{k=1,\ldots,n} (a_{jk} - a_{ik}) \rceil$$

$$u_{ij} := \lfloor \max_{k=1,\ldots,n} (a_{jk} - a_{ik}) \rfloor$$

$$d_{ij} := u_{ij} - l_{ij} + 1.$$

if $d_{ij} \leq 0$, then no integer images exist. Stop.

end

end

Set

$$D_i := \prod_{\substack{j=1\\ j\neq i}}^{m} d_{ij}.$$

Find

$$D_p = \min_{i=1,\ldots,m} D_i.$$

Set $D := D_p$ and for all $j \in M$, set $l_j := l_{pj}$, $u_j := u_{pj}$ and

$$A' = (a'_{ij}) = \begin{cases} a_{pj} & \text{if } i = 1, \\ a_{1j} & \text{if } i = p, \\ a_{ij} & \text{otherwise.} \end{cases}$$

The above subroutine finds the row which will generate the smallest D by setting $b_i = 0$ for each $i \in M$. It also checks if the matrix satisfied the necessary condition from Proposition 6.5.1.

Note that the main loop of Subroutine 2 is repeated mn times which is linear in the size of the input. Since the value of D could be significantly reduced in the case of large matrices, it may be desirable to use this subroutine when finding the integer image in this instance.

Now by combining Algorithm 4 and Subroutine 2, we can obtain the following modified algorithm.

Algorithm 5.

Input: A matrix $A \in \mathbb{R}^{m\times n}$.

Output: A vector $b \in \mathbb{Z}^n$ for which $b \in IIm(A)$ or an indication that such a b does not exist.

Run Subroutine 2.

for $i = 1$ to D

begin

if $b^{(i)} = A' \otimes (A'^* \otimes' b^{(i)})$,

Integer image exist and b is an integer image of A where

$$b := (b_j) = \begin{cases} b_p^{(i)} & \text{if } j = 1, \\ b_1^{(i)} & \text{if } j = p, \\ b_j^{(i)} & \text{otherwise.} \end{cases}$$

Stop.

else

Set $b^{(i)} := b^{(i-1)}$ and $b_m^{(i)} := b_m^{(i-1)} + 1$.

for $p = m$ to 2 do

begin

if $b_p^{(i)} > u_p$,

Set $b_p^{(i)} := l_p$ and $b_{p-1}^{(i)} := b_{p-1}^{(i)} + 1$.

end

end

6.6 Summary

In this chapter, we have investigated the integer linear system problem. We have found that the problem is relatively easy to solve for the case when the matrix A only consists of one or two columns, i.e. $n = 1$ and $n = 2$. For both of these cases, we have obtained a necessary and sufficient condition such that the matrix has an integer image.

We have also shown that if we transform a square matrix into a strongly definite matrix, its image set will not be affected. It will also be the case for its simple image set if the square matrix is strongly regular. It turns out that when A is a strongly regular and strongly definite matrix, it is not difficult to obtain the integer simple set of this matrix. We found that the integer simple image set is exactly the set of integer eigenvectors of another matrix, namely $\widetilde{C}$ which is easily obtained from A. Using the results we obtained for strongly regular matrices, we have found some sufficient conditions for an integer image to exist for any strongly definite matrix.

Finally, we have investigated the general case. We have obtained an upper and lower bound between any two components for any integer image of a finite matrix. Using this fact, we have developed an algorithm to generate all possible candidates for an integer image. Unfortunately, the complexity of the algorithm is exponential, so we have created a subroutine to minimize the maximum number of possible candidate. This algorithm provides us a benchmark on solving inter linear system for any matrices in general.

Chapter 7

On Permuted Linear Systems

7.1 Introduction

In the steady state problem, the manufacturers may have a set of starting times in which they desire to use for their system. These starting times may be chosen for various reasons; it includes the power consumption of the machines, manpower available at different times of a day, etc. But it may also be the case that the manufacturers have not assigned specified starting time for each machine. Then the problem which will arise from this situation will be to find a one to one correspondence between the starting time and the machine subject to the system will achieve steady state immediately.

Another possible scenario will be that the manufacturers have already decided a list of completion times for their products. This is likely to be the case since the manufacturers need to deliver their products to their buyers within some certain deadlines. But it may also be the case that the completion time is not restricted on a certain product; i.e. there is a freedom on choosing which products will meet which completion time as long as the completion time for all the products match the set of times specified at the start. One example in which this would happen is when all the finished products are the same but the manufacturers need to ship the products to different buyers at different times.

Finding the one to one correspondence between the specified time and machines can be modelled as a permutation problem. Therefore the two problems we discussed above can be formulated as a *permuted eigenvector problem* and *permuted linear system problem* respectively. In Chapter 3, we know that the steady state problem can be transformed into linear system problem and therefore we will only consider the permuted linear system problem in this chapter.

From [16], we also know that the permuted linear system problem is NP-complete but we will show that we can solve the problem efficiently when the matrix is small. We will develop polynomial algorithms for solving this problem for the case when $n = 2$ and $n = 3$. Then using these results, we will modify the algorithms to solve the general case.

We will assume that the vector of starting time given by the manufacturer is finite and by the same argument discussed in Chapter 2 regarding system of linear equations, we can assume without loss of generality that all matrices considered are doubly $\mathbb{R}$-astics.

We will first start by formally defining the permuted linear system problem.

Definition 7.1.1. Let $A = (a_{ij}) \in \overline{\mathbb{R}}^{m\times n}$, $b \in \mathbb{R}^m$ and $\pi \in P_m$ be a permutation, then $A(\pi)$ is the matrix in which the row indices have been permuted by π, i.e. $A(\pi) = (a_{\pi(i),j})$ and $b(\pi)$ is the vector in which the permutation π have been applied to b, i.e.

$$b(\pi) = (b_{\pi(i)}) = (b_{\pi(1)}, ..., b_{\pi(m)}).$$

Example 7.1.1. Let $b = (1, 2, 3, 4, 5)^T$ and $\pi = (1\ 3)$. Then $b(\pi) = (3, 2, 1, 4, 5)^T$ where the first element and third element in b are swapped with each other.

Problem 8. Given a matrix $A \in \overline{\mathbb{R}}^{m\times n}$ and a vector $b \in \mathbb{R}^m$, is there a $\pi \in P_m$ s.t.

$$b(\pi) = A \otimes x, \text{ for some } x \in \overline{\mathbb{R}}^n$$

7.2 Deciding whether a Permuted Vector is in the Image Set

Let us start by the case of a one column matrix. It turns out that the case when $n = 1$ is trivial. When $n = 1$ the matrix A is simply a vector. Therefore to check if b can be permuted such that it is an image of A, we can sort A and b into same ordering and check if b is a multiple of A. This method will be significant when we move on to the second half of this chapter.

7.2.1 The Case of Two Columns Matrix

Let us recall Proposition 6.3.1 in Chapter 6. As a reminder, it will be displayed again here.

Proposition 7.2.1. [23][62] Let $A \in \overline{\mathbb{R}}^{m\times 2}$ be doubly $\mathbb{R}$-astic, $m > 1$, then there exist $s, t \in M$, $s \neq t$ such that $s \in M_1(A, b)$, $t \in M_2(A, b)$ for any $b \in Im(A)$. Furthermore, s and t are the indices such that

$$a_{s1} - a_{s2} = \max_{i=1,\ldots,m} (a_{i1} - a_{i2}),$$

$$a_{t1} - a_{t2} = \min_{i=1,\ldots,m} (a_{i1} - a_{i2}).$$

Using the property from the proposition above, we can define a necessary condition for $b(\pi) \in Im(A)$.

By Proposition 7.2.1, we know that we can always find $s \in M$ such that $s \in M_1(A, b)$, $\forall b \in Im(A)$ or equivalently $\forall i \in M$,

$$a_{s1} - b_s \geq a_{i1} - b_i.$$

Now if $b(\pi) \in Im(A)$, it immediately follows that $\forall i \in M$,

$$a_{s1} - b_{\pi(s)} \geq a_{i1} - b_{\pi(i)}. \tag{7.1}$$

By using the same argument, we will get

$$a_{t2} - b_{\pi(t)} \geq a_{i2} - b_{\pi(i)}. \tag{7.2}$$

But we also know that if $b(\pi) \in Im(A)$ then $\forall i \in M$, either $i \in M_1(A, b(\pi))$ or $i \in M_2(A, b(\pi))$, i.e.

$$a_{s1} - b_{\pi(s)} = a_{i1} - b_{\pi(i)} \text{ or} \tag{7.3}$$

$$a_{t2} - b_{\pi(t)} = a_{i2} - b_{\pi(i)}. \tag{7.4}$$

Note that if $b(\pi)$ satisfies (7.1),(7.2),(7.3) and (7.4) for all $i \in M$, then by Corollary 2.3.1 it immediately implies that $b(\pi) \in Im(A)$. Therefore we can use these conditions to devise a method for deciding whether for a $b \in \mathbb{R}^m$, there exists a $\pi \in P_m$ such that $b(\pi) \in Im(A)$.

Let us consider the two matrices $B^{(1)} = (a_{i1} - b_j)$ and $B^{(2)} = (a_{i2} - b_j)$ where $b \in \mathbb{R}^m$. From these two matrices, we can see that if $b(\pi) \in Im(A)$ we can find a permutation such that it satisfies both (7.1) in $B^{(1)}$ and (7.2) in $B^{(2)}$.

Using Proposition 7.2.1 again, we know that the indices such that $s \in M_1(A, b(\pi))$ and $t \in M_2(A, b(\pi))$ when $b(\pi) \in Im(A)$ or equivalently the row s and t are the column maximum for column one and column two of the normalized matrix $\bar{A} = (a_{ij} - b_{\pi(i)})$ respectively. We also know that $s \neq t$ and therefore we can obtain $m(m-1)$ possible pairs of values and use them to find out if a permutation exists such that it satisfies (7.1) and (7.2).

This can be done by generating a $0-1$ matrix which is a matrix consisting of only zeroes and ones. We will generate this matrix for each pair of values mentioned above. For each pair of values, we first create a $m \times m$ matrix, namely E with all entries equal to ones. Then we will check from $B^{(1)}$ and $B^{(2)}$ to find the entries such that (7.3) and (7.4) are satisfied and in the possible case set the corresponding entries in E to zeroes.

Next we check again from $B^{(1)}$ and $B^{(2)}$ to find the entries such that (7.1) and (7.2) is not satisfied and set the corresponding entries in E back to one.

At the end the zeroes entries in the updated matrix imply that the corresponding entries in $B^{(1)}$ and $B^{(2)}$ will satisfy all four conditions, i.e. (7.1),(7.2), (7.3) and (7.4). If we can find m zeroes in E such that no two are in the same row or column (*independent zeroes*) then there exists a permutation π such that $b(\pi) \in Im(A)$ and the corresponding permutation from E will be π.

If we cannot find m independent zeroes for all possible $m(m-1)$ pairs of column maxima then it implies that no such permutation exists, i.e. $b(\pi) \notin Im(A) \;\; \forall \pi \in P_m$.

Note that the problem of finding m independent zeroes in the matrix E called the *bottleneck assignment problem* (BAP). Recall that BAP is the following:

Problem 9. [13] Given an $m \times m$ matrix $A = (a_{ij})$ with entries from $\mathbb{R} \cup \{\infty\}$, find $\pi \in P_m$ such that

$$\max_{i \in M} a_{i,\pi(i)} \rightarrow \min .$$

There is a number of efficient algorithms for solving this problem including one of computational complexity $O(m^{2.5})$ [54].

Algorithm 6.

Input: A matrix $A \in \mathbb{R}^{m \times 2}$ and a vector $b \in \mathbb{R}^m$.

Output: $b(\pi) \in Im(A)$ and a permutation π or $b(\pi) \notin Im(A), \forall \pi \in P_m$.

Find k s.t.

$$a_{k2} - a_{k1} = \min_i (a_{i2} - a_{i1})$$

and l such that

$$a_{l1} - a_{l2} = \min_i (a_{i1} - a_{i2}).$$

Set $B^{(1)} := (a_{i1} - b_j)$ and $B^{(2)} := (a_{i2} - b_j)$.

For $i = 1$ to m,

For $j = 1$ to m, $j \neq i$,

Set E to be the $m \times m$ all one matrix.

For $r = 1$ to m,

For $s = 1$ to m,

If $b^{(1)}_{rs} = b^{(1)}_{ki}$ or $b^{(2)}_{rs} = b^{(2)}_{lj}$,

Set $e_{rs} := 0$.

If $b^{(1)}_{rs} > b^{(1)}_{ki}$ or $b^{(2)}_{rs} > b^{(2)}_{lj}$,

Set $e_{rs} := 1$.

If π solves BAP and the solution is 0, then $b(\pi) \in Im(A)$. End of Algorithm.

Cannot find m independent 0 in E for all i and j therefore $b(\pi) \notin Im(A)$, $\forall \pi \in P_m$.

7.2.2 Computational Complexity of Algorithm 6

Now let us consider the total number of operations required for Algorithm 6 by considering each step in the algorithm. The first step will require us to compute the two rows k and l. We can see that it require $2m$ operations to find k and $2m$ operations to find l.

The next step will require us to find the two matrices $B^{(1)}$ and $B^{(2)}$ and each matrix required m^2 operations therefore we would have done $2m^2$ operations.

Now we will need to find all possible values of column maxima by using entries in b. This is represented by the two loops i, j and we will have m^2 sub-cases.

Next we will need to check through $B^{(1)}$ and $B^{(2)}$ and find out which entries satisfy the two conditions and store these entries in a matrix E. This will require six operations to obtain each entry of E and hence the number of operations required will be $6m^2$.

Finally we will need to check if E have m independent zeroes and this required $O(m^{2.5})$ operations. Therefore the total number of operations required for this algorithm is:

$$2m + 2m^2 + (m^2)(6m^2 + m^{2.5}) \sim O(m^{4.5})$$

7.2.3 The case when $n = 3$

Next we will consider the case when $n = 3$. Note that the statement of Proposition 7.2.1 cannot be extended to matrices with three or more columns. For instance let

$$A = \begin{pmatrix} 1 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

then both $b_1 = (0, 0)^T$ and $b_2 = (2, 0)^T$ are in $Im(A)$. However $M_1(A, b_1) = \{1\}$ and $M_1(A, b_2) = \{2\}$.

Since the property in Proposition 7.2.1 cannot be generalized to the case $n = 3$, therefore we do not know where the column maxima will be for any $b \in Im(A)$.

Suppose that $b \in Im(A)$ and we will again let $\bar{A} = (a_{ij} - b_i)$ be the normalized matrix of A. We know that there are m^3 different possible cases on where the column maxima will be in $\bar{A}$ (m possibilities in each column). Therefore we can consider every case and for each case, we can use the same method developed in Subsection 7.2.1 but instead we will have three matrices $B^{(1)}$, $B^{(2)}$ and $B^{(3)}$. After obtaining these three matrices, we will have to check the sufficient conditions using them.

By modifying Algorithm 6 we can obtain the following algorithm which can solve Problem 8 for the case $n = 3$.

Algorithm 7.

Input: A matrix $A \in \mathbb{R}^{m\times 3}$, and a vector $b \in \mathbb{R}^m$

Output: $b(\pi) \in Im(A)$ and a permutation π or $b(\pi) \notin Im(A), \forall \pi \in P_m$

Set $B^{(1)} := (a_{i1} - b_j)$, $B^{(2)} := (a_{i2} - b_j)$ and $B^{(3)} := (a_{i3} - b_j)$

For $u = 1$ to m

  For $v = 1$ to m

    For $w = 1$ to m

      For $i = 1$ to m

        For $j = 1$ to m

          For $k = 1$ to m

            Set E to be a $m \times m$ matrix with all entries are ones

            For $r = 1$ to m

              For $s = 1$ to m

                If $b^{(1)}_{rs} = b^{(1)}_{ui}$ or $b^{(2)}_{rs} = b^{(2)}_{vj}$ or $b^{(3)}_{rs} = b^{(3)}_{wk}$

                  Set $e_{rs} := 0$

                If $b^{(1)}_{rs} > b^{(1)}_{ui}$ or $b^{(2)}_{rs} > b^{(2)}_{vj}$ or $b^{(3)}_{rs} > b^{(3)}_{wk}$

                  Set $e_{rs} := 1$

            If π solves BAP for E and the solution is 0, then

            $b(\pi) \in Im(A)$. End of Algorithm.

Cannot find m independent 0 in E for all i and j therefore $b(\pi) \notin Im(A)\ \ \forall \pi$.

7.2.4 Computational Complexity of Algorithm 7

We can check the total number of operations required for Algorithm 7 by considering each step in the algorithm. First we will need to create three matrices $B^{(1)}$, $B^{(2)}$ and $B^{(3)}$ and each matrix required m^2 operations therefore we would have done $3m^2$ operations.

Next we will consider the three loops: u, v and w in the algorithm. These represent all the cases on where the column maxima will be for each column of A and there are m different possibilities in each column. Hence we will have m^3 cases.

Then for each different case we will need to find all possible values of column maxima by using entries in b. This is represented by the three loops i, j and k and we will have m^3 sub-cases.

For each sub case we will need to check through $B^{(1)}$, $B^{(2)}$ and $B^{(3)}$ and find out which entries satisfy the two conditions and store these entries in a matrix E. This will require 8 operations to obtain each entry of E and hence the number of operations required will be $8m^2$.

Finally we will need to check if E have m independent zeroes and this can be checked with $m^{2.5}$ operations. Therefore the total number of operations required for this algorithm is:

$$3m^2 + (m^3)(m^3)(8m^2 + m^{2.5}) \sim O(m^{8.5})$$

7.2.5 The case when $n > 3$

The method developed for the case when $n = 3$ can be generalized for the case when $n > 3$. We formally describe it here although it is computationally infeasible since the number of operations increase exponentially. For the case when $n > 3$ we will have:
n matrices $B^{(1)}, ..., B^{(n)}$
m^n cases on where the column maxima will be
m^n sub-cases on the value of column maxima
$(2n + 2)m^2$ operations for obtaining E
$m^{2.5}$ operations for checking if E have independent 0.

Therefore for $n > 3$ this method will require:

$$nm^2 + (m^n)(m^n)((2n + 2)m^2 + m^{2.5}) \sim O(nm^{(2n+2.5)}) \text{ operations.}$$

We can see that when n increases the computational complexity increases exponentially.

7.3 Finding the Permuted Vector Closest to the Image Set

In general, it is possible that a given vector b cannot be permuted into an image of A. One of the question the manufacturer may want to ask is if we can find a permutation on the vector b such that it is close to being the image A. First we will need to define the distance of a vector from the image set by the following definition:

Definition 7.3.1. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and $b = (b_1, ..., b_m) \in \mathbb{R}^m$. Suppose that $\bar{x} = (\bar{x}_1, ..., \bar{x}_n) \in \overline{\mathbb{R}}^n$ is the principal solution to the system $A \otimes x = b$, that is $\bar{x} = A^* \otimes' b$. Then we will define

$$slk(A, b) = \max_{i\in M} |(A \otimes \bar{x})_i - b_i|$$

to be the *slack* of b from the image of A. The value of the slack is called the *Chebyshev-norm*.

Then the problem can be formulated as below:

Problem 10. Given a matrix $A \in \overline{\mathbb{R}}^{m\times n}$ and a vector $b = (b_1, ..., b_m) \in \mathbb{R}^m$, find a permutation $\pi \in P_m$ that minimizes

$$slk(A, b(\pi)) = \max_{i\in M} |(A \otimes (A^* \otimes' b(\pi)))_i - b_{\pi(i)}|.$$

Note that if we can find a permutation π such that $slk(A, b(\pi)) = 0$, i.e. there is no slack between $b(\pi)$ and the image of A then it means that $b(\pi) \in Im(A)$ and then Problem 8 is solved.

7.3.1 The One Column Problem

Since finding if a permuted vector is in any image set is NP-complete and the problem of finding the permuted vector closest to the image set is more complex, we do not expect an efficient method (a method which solves the problem in polynomial time) in general but will do so for the case when $n = 1, 2$. We may assume without loss of generality that the entries are sorted.

Proposition 7.3.1. Suppose that $A, b \in \mathbb{R}^m$ and A is ordered non-decreasingly, i.e.

$$a_1 \leq a_2 \leq ... \leq a_m$$

Then the permutation π solves Problem 10 if $b(\pi)$ is also ordered
non-decreasingly, i.e.

$$b_1(\pi) \leq b_2(\pi) \leq ... \leq b_m(\pi)$$

Proof. Let $A, b \in \mathbb{R}^m$ and suppose that b is already permuted s.t. it is a solution to Problem 10. We have

$$slk(A, b) = \max_{i=1,...,m} |(A \otimes \bar{x})_i - b_i|$$

and

$$\begin{aligned} \bar{x} &= \min_{i=1,...,m} (-a_i + b_i) \\ &= -\max_{i=1,...,m} (a_i - b_i). \end{aligned}$$

Therefore

$$\begin{aligned} slk(A,b) &= \max_{j=1,\ldots,m} |(A \otimes (-\max_{i=1,\ldots,m}(a_i - b_i))_j - b_j| \\ &= \max_{j=1,\ldots,m} |(a_j - \max_{i=1,\ldots,m}(a_i - b_i)) - b_j| \\ &= \max_{j=1,\ldots,m} |(a_j - b_j) - \max_{i=1,\ldots,m}(a_i - b_i)| \\ &= |\min_{j=1,\ldots,m}(a_j - b_j) - \max_{i=1,\ldots,m}(a_i - b_i)| \\ &= \max_{j=1,\ldots,m}(a_j - b_j)) - \min_{i=1,\ldots,m}(a_i - b_i). \end{aligned}$$

For simplicity we will denote

$$\Delta = \max_{i=1,\ldots,m}(a_i - b_i)$$

and

$$\delta = \min_{i=1,\ldots,m}(a_i - b_i).$$

Therefore

$$\Delta > a_i - b_i \geq \delta \quad \forall i$$

and

$$slk(A,b) = \Delta - \delta.$$

Suppose that b is not ordered non-decreasingly, that is $\exists k$ s.t. $b_k > b_{k+1}$. Let b' be the vector

$$\begin{aligned} b'_k &= b_{k+1} \\ b'_{k+1} &= b_k \\ b'_i &= b_i, \quad i \neq k, k+1 \end{aligned}$$

We know that

$$\Delta \geq a_k - b_k \geq \delta \tag{7.5}$$

and

$$\Delta \geq a_{k+1} - b_{k+1} \geq \delta \tag{7.6}$$

So if we consider swapping the k and $k+1$ entries of the vector b then using the fact that $b_{k+1} - b_k < 0$ and 7.5 we will get

$$(a_k - b_{k+1}) > (a_k - b_{k+1}) + (b_{k+1} - b_k) = (a_k - b_k) \geq \delta. \tag{7.7}$$

Then we will use the fact that A is ordered, i.e. $a_{k+1} - a_k \geq 0$ and 7.6 then we will get the following inequality:

$$(a_k - b_{k+1}) \leq (a_k - b_{k+1}) + (a_{k+1} - a_k) = (a_{k+1} - b_{k+1}) \leq \Delta \tag{7.8}$$

Combining 7.7 and 7.8 we will get

$$\Delta \geq a_k - b_{k+1} > \delta \tag{7.9}$$

We can use a similar method to obtain the following two inequalities:

$$(a_{k+1} - b_k) \geq (a_{k+1} - b_k) + (a_k - a_{k+1}) = (a_k - b_k) \geq \delta \tag{7.10}$$

$$(a_{k+1} - b_k) < (a_{k+1} - b_k) + (b_k - b_{k+1}) = (a_{k+1} - b_{k+1}) \leq \Delta. \tag{7.11}$$

Combining 7.10 and 7.11 we will get

$$\Delta > a_{k+1} - b_k \geq \delta \tag{7.12}$$

Therefore using 7.3.1, 7.9 and 7.12 we know that

$$\Delta \geq a_i - b_i \geq \delta \ \ \forall i \neq k, k+1$$

$$\Delta \geq a_k - b_{k+1} > \delta$$

$$\Delta > a_{k+1} - b_k \geq \delta$$

and hence $\Delta \geq a_i - b'_i \geq \delta \ \ \forall i.$

This implies that $slk(A, b') \leq slk(A, b)$. Since b is already a solution to Problem 10 before the permutation therefore b' is also a solution to Problem 10. We can then continue to swap any two unordered elements of b and after a finite number of swaps the vector will become ordered and the resulting vector will still be a solution to Problem 10 and hence the statement. □

Corollary 7.3.1. Suppose that $A, b \in \mathbb{R}^m$ and $\pi, \sigma \in P_m$ then the permutation $\sigma^{-1}\pi\sigma$ solves Problem 10 if $A(\sigma)$ is ordered and $b(\pi\sigma)$ is ordered.

Proof. Let $A, b \in \mathbb{R}^m$ and let σ to be the permutation such that A is ordered. If we apply the permutation σ to both A and b the problem will remain the same but now $A(\sigma)$ will be ordered then we can apply Proposition 7.3.1 and we will get $\pi\sigma$ for the solution. Finally we could apply the inverse permutation of σ to get b for the original A. □

7.3.2 The Two Columns Problem

Now we have considered the one column problem. The next step will be to consider two columns problem. First we will develop a method on transforming n columns problem into a one column problem by using the following proposition.

Proposition 7.3.2. Suppose that $A \in \overline{\mathbb{R}}^{m\times n}$ and $b \in \mathbb{R}^m$ then

$$\begin{array}{lcl} \pi \text{ is a solution} & \Longrightarrow & \exists\, x \in \mathbb{R}^n \text{ s.t. } \pi \text{ is a solution to the one column} \\ \text{to Problem 10} & & \text{problem with the matrix of the system } A' = A \otimes x \end{array}$$

Proof. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astics, $b \in \mathbb{R}^m$ and π be a solution to Problem 10 then

$$slk(A, b(\pi)) \text{ is minimized} \Longrightarrow \max_{i=1,\dots,m} |(A \otimes (A \otimes' b(\pi)))_i - b(\pi)_i| \text{ is minimized.}$$

Let $x = A \otimes' b(\pi) \in \mathbb{R}^n$ and $A' = A \otimes x \in \mathbb{R}^m$ then we will have

$$\max_{i=1,\dots,m} |(A')_i - b(\pi)_i| \text{ is minimized} \Longrightarrow slk(A', b(\pi)) \text{ is minimized}$$

Therefore π is a solution to Problem 10 for the one column problem A' with $x = A \otimes' b(\pi)$ $\square$

Unfortunately the above result will require us to know π first in order to find out x and transform A into a one column problem. But if we only consider $n = 2$ case, i.e. let $A \in \overline{\mathbb{R}}^{m\times 2}$ and $x = (x_1, x_2)^T \in \mathbb{R}^2$ then the images of A can be written as follows:

$$\begin{aligned} A \otimes x =& (A_1, A_2) \otimes (x_1, x_2)^T \\ &= x_1 \otimes A_1 \oplus x_2 \otimes A_2 \\ &= x_1 \otimes (A_1 \oplus (x_2 \otimes x_1^{-1}) \otimes A_2). \end{aligned}$$

Note that if we fix $x_1 = 0$ and consider all possible values of x_2 then we will obtain a set of vectors that are in the images of A and the multiples of these vectors will be the whole image set for A. Using this set of vectors we will know how the image set looks like and therefore we can use this property to transform the two columns problem into a one column problem.

Now if we obtain all possible permutations π which solve the one column problem from the set of vectors then from Proposition 7.3.2 we know that one of these permutations will be a solution to the two columns problem. So we can check the slack on A for each of these permutations and the permutation which gives the minimal slack will be a solution to the two columns problem. We will illustrate the method on solving the two columns problem by using the following example.

Example 7.3.1. Given $A = \begin{pmatrix} 1 & 1 \\ 2 & -6 \\ 3 & -11 \\ 4 & -16 \\ 5 & -21 \end{pmatrix}$ and $b = \begin{pmatrix} 1 \\ 8 \\ 9 \\ 14 \\ 20 \end{pmatrix}$.
Find a permutation π s.t. $slk(A, b(\pi))$ is minimized.

The first step will be considering the image set of A by fixing x_1 to be zero and we will have

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \oplus (x_2 \otimes \begin{pmatrix} 1 \\ -6 \\ -11 \\ -16 \\ -21 \end{pmatrix})$$

If we assume x_2 to be extremely small, i.e. $x_2 = \epsilon$, then the second column will be ignored and we will have the first column as our first vector. Since the first column of A and b are already ordered then the permutation that solved the one column problem for our first vector will the identity, i.e. $\pi_1 = id$.

The next step will be to increase x_2. Note that the slack may have changed as x_2 increases but this will not affect the permutation until the image is not ordered anymore. Now we will keep increasing x_2 until it reaches the point when an entry on the second column starts to affect the image for the first time.

If we consider the minimum difference between the corresponding entry on the first and second column, i.e.

$$\min(1-1, 2-(-6), 3-(-11), 4-(-16), 5-(-21)) = 0.$$

Then we know that when $x_2 = 0$ the first entry of the second column will be the first to affect the images. We will call this entry *active*. When an entry becomes active, it means that this entry will continue to change in value as x_2 increases. Therefore the resulting image will also change as x_2 increases. Henceforth we will take a note of this by first create an empty set called S. Then we will include the index for this entry into S, i.e. $S = \{1\}$ in this case. Since the order is not affected so we will start increasing x_2 again.

Now we have increased x_2 to 1 then

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \oplus (1 \otimes \begin{pmatrix} 1 \\ -6 \\ -11 \\ -16 \\ -21 \end{pmatrix}) = \begin{pmatrix} 2 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}.$$

We can see that the first and the second entry of the image is now equal and the image will not be ordered as x_2 increases beyond this point. Therefore we will have a new permutation and using Corollary 7.3.1 we have $\sigma = (1\ 2)$ and $\pi = (1\ 2)$, hence $\pi_2 = \sigma^{-1}\pi\sigma = (1\ 2)$. For simplicity we would want to keep the image ordered therefore we will apply σ to both A and b and increase x_2 by using the permuted A.

Once again we will increase x_2 and when $x_2 = 2$ we will have

$$\begin{pmatrix} 2 \\ 1 \\ 3 \\ 4 \\ 5 \end{pmatrix} \oplus (1 \otimes \begin{pmatrix} -6 \\ 1 \\ -11 \\ -16 \\ -21 \end{pmatrix}) = \begin{pmatrix} 2 \\ 3 \\ 3 \\ 4 \\ 5 \end{pmatrix}.$$

This time the second and the third entry of the image is now equal and again the image will not be ordered as x_2 increases beyond this point. Therefore we will have another permutation and using Corollary 7.3.1 again, we will have $\sigma = (1\ 3\ 2)$ and $\pi = (2\ 3\ 1)$, hence $\pi_3 = \sigma^{-1}\pi\sigma = (1\,2\,3)$. Now we will apply σ to A and repeat this process again.

We will continue to get new permutation in every step until $x_2 = 8$. Then the second entry of the second column will start to affect the images and this entry will now become active. Then we will have a new element in S and it is now $S = \{1, 2\}$. By repeating this process we will have obtain new permutations until all entries become active. Table 7.1 and 7.2 represent our results.

Step	x_2	A_1	A_2	$x_2 \otimes A_2$	$A_1 \oplus x_2 \otimes A_2$	$b(\pi)$	$A(\sigma)$	
1	$-\infty$	1	1	$-\infty$	1	1	1	1
		2	-6	$-\infty$	2	8	2	-6
		3	-11	$-\infty$	3	9	3	-11
		4	-16	$-\infty$	4	14	4	-16
		5	-21	$-\infty$	5	20	5	-21
2	0	1	1	1	**1**	1	1	1
		2	-6	-6	2	8	2	-6
		3	-11	-11	3	9	3	-11
		4	-16	-16	4	14	4	-16
		5	-21	-21	5	20	5	-21
3	1	1	1	2	**2**	8	2	-6
		2	-6	-5	2	1	1	1
		3	-11	-10	3	9	3	-11
		4	-16	-15	4	14	4	-16
		5	-21	-20	5	20	5	-21

Step	x_2	A_1	A_2	$x_2 \otimes A_2$	$A_1 \oplus x_2 \otimes A_2$	$b(\pi)$	$A(\sigma)$	
4	2	2	−6	−4	2	9	2	−6
		1	1	3	**3**	1	3	−11
		3	−11	−9	3	8	1	1
		4	−16	−14	4	14	4	−16
		5	−21	−19	5	20	5	−21
5	3	2	−6	−3	2	14	2	−6
		3	−11	−8	3	1	3	−11
		1	1	4	**4**	8	4	−16
		4	−16	−13	4	9	1	1
		5	−21	−18	5	20	5	−21
6	4	2	−6	−2	2	20	2	−6
		3	−11	−7	3	1	3	−11
		4	−16	−12	4	8	4	−16
		1	1	5	**5**	9	5	−21
		5	−21	−17	5	14	1	1
7	8	2	−6	2	**2**	20	2	−6
		3	−11	−3	3	1	3	−11
		4	−16	−8	4	8	4	−16
		5	−21	−13	5	9	5	−21
		1	1	9	**9**	14	1	1
8	9	2	−6	3	**3**	20	3	−11
		3	−11	−2	3	8	2	−6
		4	−16	−7	4	1	4	−16
		5	−21	−12	5	9	5	−21
		1	1	10	**10**	14	1	1
9	10	3	−11	−1	3	20	3	−11
		2	−6	4	**4**	9	4	−16
		4	−16	−6	4	1	2	−6
		5	−21	−11	5	8	5	−21
		1	1	11	**11**	14	1	1
10	11	3	−11	0	3	20	3	−11
		4	−16	−5	4	14	4	−16
		2	−6	5	**5**	1	5	−21
		5	−21	−10	5	8	2	−6
		1	1	12	**12**	9	1	1
11	14	3	−11	3	**3**	20	3	−11
		4	−16	−2	4	14	4	−16
		5	−21	−7	5	1	5	−21
		2	−6	8	**8**	8	2	−6
		1	1	15	**15**	9	1	1
12	15	3	−11	4	**4**	20	4	−16
		4	−16	−1	4	14	3	−11
		5	−21	−6	5	8	5	−21
		2	−6	9	**9**	1	2	−6
		1	1	16	**16**	9	1	1
13	16	4	−16	0	4	20	4	−16
		3	−11	5	**5**	14	5	−21
		5	−21	−5	5	9	3	−11
		2	−6	10	**10**	1	2	−6
		1	1	17	**17**	8	1	1

Step	x_2	A_1	A_2	$x_2 \otimes A_2$	$A_1 \oplus x_2 \otimes A_2$	$b(\pi)$	$A(\sigma)$	
		4	-16	4	**4**	20	4	-16
		5	-21	-1	5	14	5	-21
14	20	3	-11	9	**9**	9	3	-11
		2	-6	14	**14**	1	2	-6
		1	1	21	**21**	8	1	1
		4	-16	5	**5**	20	5	-21
		5	-21	0	5	14	4	-16
15	21	3	-11	10	**10**	9	3	-11
		2	-6	15	**15**	8	2	-6
		1	1	22	**22**	1	1	1
		5	-21	5	**5**	20	5	-21
		4	-16	10	**10**	14	4	-16
16	26	3	-11	15	**15**	9	3	-11
		2	-6	20	**20**	8	2	-6
		1	1	27	**27**	1	1	1

Table 7.1: The results obtained when the value for x_2 increase continuously.

Step	x_2	Permutation	Comment
1	$-\infty$	id	$\pi_1 = id$, $S = \phi$
2	0	id	First entry becomes active, $S = \{1\}$
3	1	$(1\ 2)$	New permutation obtained $\pi_2 = (1\ 2)$
4	2	$(1\ 2\ 3)$	New permutation obtained $\pi_3 = (1\ 2\ 3)$
5	3	$(1\ 2\ 3\ 4)$	New permutation obtained $\pi_4 = (1\ 2\ 3\ 4)$
6	4	$(1\ 2\ 3\ 4\ 5)$	New permutation obtained $\pi_5 = (1\ 2\ 3\ 4\ 5)$
7	8	$(1\ 2\ 3\ 4\ 5)$	Second entry becomes active, $S = \{1, 2\}$
8	9	$(1\ 3\ 4\ 5)$	New permutation obtained $\pi_6 = (1\ 3\ 4\ 5)$
9	10	$(1\ 3\ 2\ 4\ 5)$	New permutation obtained $\pi_7 = (1\ 3\ 2\ 4\ 5)$
10	11	$(1\ 3\ 5)(2\ 4)$	New permutation obtained $\pi_8 = (1\ 3\ 5)(2\ 4)$
11	14	$(1\ 3\ 5)(2\ 4)$	Third entry becomes active, $S = \{1, 2, 3\}$
12	15	$(1\ 4\ 2\ 3\ 5)$	New permutation obtained $\pi_9 = (1\ 4\ 2\ 3\ 5)$
13	16	$(1\ 4\ 2\ 5)$	New permutation obtained $\pi_{10} = (1\ 4\ 2\ 5)$
14	20	$(1\ 4\ 2\ 5)$	Fourth entry becomes active, $S = \{1, 2, 3, 4\}$
15	21	$(1\ 5)(2\ 4)$	New permutation obtained $\pi_{11} = (1\ 5)(2\ 4)$
16	26	$(1\ 5)(2\ 4)$	Fifth entry becomes active, $S = \{1, 2, 3, 4, 5\}$

Table 7.2: Summary on the results obtained.

Therefore after step sixteen all entries will become active. The order will not change as x_2 increases beyond this point and the resulting image will become a multiple of the second column of A. Hence all permutations are being obtained and we will need to compare all eleven permutations and find out which one will give out the best slack.

i	π_i	$b(\pi_i)$	$A \otimes (A^* \otimes' b(\pi_i))$	$slk(A, b(\pi_i))$
1	id	1 8 9 14 20	1 2 3 4 5	15
2	(1 2)	8 1 9 14 20	8 1 2 3 4	16
3	(1 2 3)	9 1 8 14 20	8 1 2 3 4	16
4	(1 2 3 4)	14 1 8 9 20	8 1 2 3 4	16
5	(1 2 3 4 5)	20 1 8 9 14	8 1 2 3 4	12
6	(1 3 4 5)	20 8 1 9 14	13 6 1 2 3	11
7	(1 3 2 4 5)	20 9 1 8 14	13 6 1 2 3	11
8	(1 3 5)(2 4)	20 14 1 8 9	13 6 1 2 3	8
9	(1 4 2 3 5)	20 14 8 1 9	18 11 6 1 2	7
10	(1 4 2 5)	20 14 9 1 8	18 11 6 1 2	6

i	π_i	$b(\pi_i)$	$A \otimes (A^* \otimes' b(\pi_i))$	$slk(A, b(\pi_i))$
11	(1 5)(2 4)	20 14 9 8 1	20 13 8 3 1	5

Table 7.3: The slacks obtained from all the possible solution.

The table above shows that the permutation which gives out the best slack is π_{11} and therefore π_{11} will be our solution to Problem 10.

From Example 7.3.1, we now have the general idea on how this method work. So we will need to formulate this method precisely and create an algorithm for solving solve Problem 10. The first step will be obtaining the set of vectors which represent the images and use it to find the permutation that is a solution to the one column problem. For simplicity we will let

$$A_1 \oplus (x_2 \otimes A_2) = c.$$

First we will start with the value when $x_2 = -\infty$, then the resulting vector c will be the first column of A. So using Corollary 7.3.1, we can find a permutation π_1 which will solve this one column problem. Hence we will get our first permutation. Then we will need to apply the permutation σ to both A and b to transform the two columns problem such that $c(\sigma)$ is ordered.

Then for the first time, we will need to find our next vector, i.e. we will start increasing the x_2 value until a value in the second column replaces the corresponding value in the first column to become an entry of c. This happens when

$$c_r = a_{r1} \oplus x_2 \otimes a_{r2} = x_2 \otimes a_{r2} \qquad \text{and}$$

$$c_i = a_{i1} \oplus x_2 \otimes a_{i2} = a_{i1} \qquad \forall i \neq r$$

i.e. when

$$x_2 = (a_{r1} - a_{r2}) = \min_{i=1,\dots,m} (a_{i1} - a_{i2}).$$

Then the r^{th} entry of the second column will become *active* and will start to affect the r^{th} entry of c. We will now add r into the set S where S is defined to be the set of indices that are active on the second column, i.e.

$$S = \{i \in M \mid a_{i1} \oplus x_2 \otimes a_{i2} = x_2 \otimes a_{i2}\}$$

We will suppose the value of x_2 has started to increase again and we will stop when the system satisfies one of the following two possible cases:

Case 1. Another entry from the second column (*inactive*) will start to replace an entry in the first column to become an entry in c, i.e. $\exists k \notin S$ such that

$$\begin{aligned} c_k &= a_{k1} \oplus x_2 \otimes a_{k2} &&= x_2 \otimes a_{k2} && \text{and} \\ c_i &= a_{i1} \oplus x_2 \otimes a_{i2} &&= a_{i1} && \forall i \notin S, i \neq k. \end{aligned}$$

This happens when

$$x_2 = (a_{k1} - a_{k2}) = \min_{i \notin S} (a_{i1} - a_{i2}).$$

Then the k^{th} entry of the second column will start to become active and we will add k into the set S.

Case 2. c is not ordered anymore, i.e. $c_l = c_{l+1}$ then the one column problem is changed and we will have a new permutation. This can only happen when the k^{th} entry is active and the $(l+1)^{st}$ entry is not active. Therefore x_2 has to be increased up to a value s.t. the following is satisfied:

$$\begin{aligned} & a_{l1} \oplus x_2 \otimes a_{l2} &&\geq a_{l+1,1} \oplus x_2 \otimes a_{l+1,2} && l \in S, \ \ l+1 \notin S \\ \Longrightarrow \quad & x_2 \otimes a_{l2} &&\geq a_{l+1,1} && l \in S, \ \ l+1 \notin S \end{aligned}$$

Then the value of x_2 will be

$$x_2 = (a_{l+1,1} - a_{l2}) = \min_{\substack{i \in S \\ i+1 \notin S}} (a_{i+1,1} - a_{i2})$$

Then the l^{th} and the $(l+1)^{st}$ entry will need to be swapped to achieve the order in c again. Therefore we can use Corollary 7.3.1 to obtain a new permutation and the new permutation will be $\pi_i \sigma_i$ where π_i is the permutation obtained before the order is changed and $\sigma_i = (l,\ l+1)$. Finally we will need to apply the permutation σ_i to both A and b to transform the two columns problem so that $c(\sigma_i)$ is ordered.

We continue to check for each one of the above cases after a new entry is active or after a new permutation is obtained until it reaches the point such that all the entries in the second column are active. This is because after the value of x_2 passes this point, the obtained vector will only be the multiple of the second column. Therefore the permutation will not change anymore after this point and we can end our algorithm here.

From Example 7.3.1 we can see that when the first entry in the second column of A becomes active, we have obtained five permutations before second entry in column two is active. Then we have obtained four permutations when the second entry becomes active and so on. So at the end we have obtained

$$1 + 4 + 3 + 2 + 1 = 11$$

permutations. In general case, the first active entry can produce at most $(m-1)$ permutations since the order of c can change at most $(m-1)$ times. Then the second active entry can produce at most $(m-2)$ permutations as the order of c can change at most $(m-2)$ times. This is due to the fact that the difference between the value at the first and second active entry for c is fixed. Similarly when the k^{th} entry become active it can produce at most $(m-k)$ permutations. Finally we need to consider the first permutation we obtained from the first column of A and hence the maximum number of permutations obtained will be

$$\begin{aligned}1+(m-1)+(m-2)+...+(m-m) &= 1+\sum_{k=1}^{m}(m-k)\\ &= 1+\sum_{k=1}^{m-1}(k)\\ &= 1+\frac{1}{2}(m)(m-1).\end{aligned}$$

After finding all the possible permutations, we will need to check the slack of b on the image of A for them. The permutation which gives the best result will be the solution to Problem 10 for the two columns problem. Note that if we use this algorithm and find a permutation π s.t. $slk(A, b(\pi)) = 0$, it means that $b(\pi) \in Im(A)$ so this algorithm may also solve Problem 8 for the $n = 2$ case.

Algorithm 8.

Input: A matrix $A \in \mathbb{R}^{m\times 2}$, and a vector $b \in \mathbb{R}^m$

Output: A permutation π and $slk(A, b(\pi))$

Find permutations σ and π_1 where $A_1(\sigma)$ and $b(\pi\sigma)$ are ordered.

Set $A := A(\sigma)$, $S := \emptyset$ and $p := 1$.

While $|S| \neq m$,

Let

$$\alpha_1 := (a_{k1} - a_{k2}) = \min_{i\notin S} (a_{i1} - a_{i2}) \quad \text{and}$$

$$\alpha_2 := (a_{l+1,1} - a_{l2}) = \min_{\substack{i\in S\\ i+1\notin S}} (a_{i+1,1} - a_{i2}).$$

If $\alpha_1 = \min(\alpha_1, \alpha_2)$,

Add k to S;

Else

Let $\pi_{p+1} := \pi_p(l, l+1)$. Set $A := A(l, l+1)$ and $p := p+1$.

Let $\pi := \pi_r$ and $slk(A, b(\pi)) := slk(A, b(\pi_r))$ where

$$slk(A, b(\pi_r)) = \min_{j=1,\ldots,p} slk(A, b(\pi_j)).$$

7.4 Summary

In this chapter, we have investigated the problem of permuted linear systems. We know that this problem is NP-complete [16]. But we have found out that the case is easily solvable for the case when the matrix consists of only one column. We also developed an algorithm on solving the case when the matrix consists of only two columns by using a result we presented in Chapter 6 (Proposition 6.3.1). The immediate consequence of this is that we have developed a similar method for solving the general case although the computational complexity will be exponential.

We have also considered the case when a given vector may not be permuted into an image of the given matrix, but we would like to find out a permutation such that the permuted vector is closest to the image set. We have shown that this problem is trivial for the one column case and using this we have developed a solution method on solving the two columns case.

Chapter 8

Heuristics for the Permuted Linear Systems Problem

8.1 Introduction

Since the permuted linear systems problem is NP-complete, we know an exact solution method is highly unlikely to be efficient. Therefore in this chapter we will develop some forms of heuristic to find a vector that is as close as possible to a permuted image of A. The distance is again measured by the Chebyshev norm.

As a reminder the definition of doubly $\mathbb{R}$-astic, the Chebyshev norm and the permuted linear systems problem will be repeated as Definition 8.1.1, Definition 8.1.2 and Problem 11, respectively.

Definition 8.1.1. [29] Let $A \in \overline{\overline{\mathbb{R}}}^{m\times n}$ be a matrix which has at least one finite entry on each row (column) then A is called *row* $\mathbb{R}$*-astic* (*column* $\mathbb{R}$*-astic*). A is called *doubly* $\mathbb{R}$*-astic* if it is both row and column $\mathbb{R}$-astic.

Definition 8.1.2. Let $A \in \overline{\mathbb{R}}^{m\times n}$ be doubly $\mathbb{R}$-astic and $b = (b_1, ..., b_m) \in \mathbb{R}^m$. Suppose that $\bar{x} = (\bar{x}_1, ..., \bar{x}_n) \in \overline{\mathbb{R}}^n$ is the principal solution to the system $A \otimes x = b$, that is $\bar{x} = A^* \otimes' b$.

Then we will define

$$slk(A, b) = \max_{i \in M} |(A \otimes \bar{x})_i - b_i|$$

to be the *slack* of b from the image of A. The value of the slack is called the *Chebyshev-norm*.

Problem 11. Given a matrix $A \in \overline{\mathbb{R}}^{m \times n}$ and a vector $b = (b_1, ..., b_m) \in \mathbb{R}^m$, find a permutation $\pi \in P_m$ that minimizes

$$slk(A, b(\pi)) = \max_{i \in M} |(A \otimes (A^* \otimes' b(\pi)))_i - b_{\pi(i)}|.$$

8.2 The Steepest Descent Method

We will first consider using what is called the *steepest descent method*. The principle idea of this method is to start with a permutation which can be randomly chosen and we evaluate the corresponding slack for this permutation. Then we will find other permutations which are often derived from the starting permutation and evaluate the corresponding slack generated from these permutations. Sometimes these permutations are said to be in the *neighborhood* of the starting permutation. We will compare these newly obtained values with the values obtained from the starting permutation. If we have found a better slack from one of these values, we say that we will found an improved solution and we will take the permutation which generated the smallest slack, i.e. the best improvement. We will continue this process until we cannot find a permutation with better slack and we will stop this process. Then the solution obtained from the previous step with be our solution.

Note that for this method, we never continue if we cannot find a better permutation at the end of a step. In Section 8.5, we will discuss a method which will continue to search for better permutations in the case when worse permutations can only be found at the end of a step.

Now we will need to devise a method which enables us to find the permutations in the neighborhood of the starting permutation. Neighborhood can be defined in various ways. In this chapter, we will define the neighborhood of a permutation by swapping two elements in the permutation.

8.2.1 Full Local Search

The following method is a combination of steepest descent and local search method. We will consider swapping two elements at one time, the full local search would mean to check swaps for every component before hand. We will use the swap that gives out the best result.

We will first need to choose a starting permutation. This starting permutation can be any randomly generated and for simplicity we can choose the identity as a starting permutation.

Then we will apply a starting permutation π on the vector b and obtain the slack from this starting permutation. After that we will check all two elements swaps and we will find one which results in the greatest decrease of slack and we will apply this swap to b and repeat the process. The method can be formulated as the following algorithm.

Algorithm 9.

Input: A matrix $A \in \mathbb{R}^{m \times n}$, a vector $b \in \mathbb{R}^m$ and a starting permutation $\pi \in P_m$.

Output: A permutation $\bar{\pi}$ and $slk(A, b(\bar{\pi}))$.

Set $z = 1$ and $\pi_z := \pi$.

Set $b := b(\pi)$ and find $slk(A, b)$.

While $slk(A, b) \neq 0$

  For $i = 1$ to m

    For $j = 1$ to m, $j \neq i$

      Find $slk(A, b(i, j))$.

Let

$$slk(A, b(r, s)) = \min_{i,j} slk(A, b(i, j)).$$

If $slk(A, b(r, s)) < slk(A, b)$

Set $\pi_{z+1} := \pi_z(r, s)$ and $z := z + 1$.

else

$slk(A, b)$ cannot be improved anymore. Therefore $\bar{\pi} := \pi_z$ and

$slk(A, b(\bar{\pi})) := slk(A, b(\pi_z))$. Stop.

$slk(A, b(\pi_z)) = 0$ implies that $b(\pi_z)$ is an image of A and therefore $\bar{\pi} := \pi_z$ and $slk(A, b(\bar{\pi}_z)) = 0$.

Example 8.2.1. Given $A = \begin{pmatrix} 1 & 1 & 3 \\ 2 & 4 & 2 \\ 4 & -1 & 5 \\ 3 & 6 & 0 \end{pmatrix}$ and $b = \begin{pmatrix} 10 \\ 5 \\ 4 \\ 1 \end{pmatrix}$.

Find a permutation π s.t. $slk(A, b(\pi))$ is minimized.

First we need to decide a starting permutation. For convenience we have used the $identity$ as the starting permutation and the resulting slack will be:

$$
\begin{aligned}
A \otimes \bar{x} &= \begin{pmatrix} 1 & 1 & 3 \\ 2 & 4 & 2 \\ 4 & -1 & 5 \\ 3 & 6 & 0 \end{pmatrix} \otimes \left(\begin{pmatrix} -1 & -2 & -4 & -3 \\ -1 & -4 & 1 & -6 \\ -3 & -2 & -5 & 0 \end{pmatrix} \otimes' \begin{pmatrix} 10 \\ 5 \\ 4 \\ 1 \end{pmatrix} \right), \\
&= \begin{pmatrix} 1 & 1 & 3 \\ 2 & 4 & 2 \\ 4 & -1 & 5 \\ 3 & 6 & 0 \end{pmatrix} \otimes \begin{pmatrix} -2 \\ -5 \\ -1 \end{pmatrix}, \\
&= \begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}.
\end{aligned}
$$

Therefore $slk(A, b) = max(10 - 2, 5 - 1, 4 - 4, 1 - 1) = 8$.

Since this permutation does not give b as an image of A, we want to find another permutation such that the permuted b is an image of A or as close to being the image as possible. So we would want to find two entries of b to swap; to find these two entries we will need to check for all possible swaps.

Swaps	New b	$A \otimes \bar{x}$	slack
(1 2)	$\begin{pmatrix} 5 \\ 10 \\ 4 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}$	9
(1 3)	$\begin{pmatrix} 4 \\ 5 \\ 10 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 4 \\ 3 \\ 6 \\ 1 \end{pmatrix}$	4

Swaps	New b	$A \otimes \bar{x}$	slack
(1 4)	$\begin{pmatrix} 1 \\ 5 \\ 4 \\ 10 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 4 \\ 4 \\ 6 \end{pmatrix}$	4
(2 3)	$\begin{pmatrix} 10 \\ 4 \\ 5 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 2 \\ 5 \\ 1 \end{pmatrix}$	7
(2 4)	$\begin{pmatrix} 10 \\ 1 \\ 4 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 3 \end{pmatrix}$	8
(3 4)	$\begin{pmatrix} 10 \\ 5 \\ 1 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 2 \\ 1 \\ 4 \end{pmatrix}$	11

Table 8.1: The First Iteration of the Full Local Search.

From Table 8.1, we noticed that if we swap the first and the third entry or the first and the fourth entry, we will obtain best slack. Therefore we will need to choose one of these two vectors to be our new b and repeat the process again. For simplicity we will use the vector $(4, 5, 10, 1)^T$ as our new b since we have obtained this vector first from the above calculation. We will get the following result:

Swaps	New b	$A \otimes \bar{x}$	slack
(1 2)	$\begin{pmatrix} 5 \\ 4 \\ 10 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 4 \\ 3 \\ 6 \\ 1 \end{pmatrix}$	4
(1 3)	$\begin{pmatrix} 10 \\ 5 \\ 4 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}$	8
(1 4)	$\begin{pmatrix} 1 \\ 5 \\ 10 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 2 \\ 4 \\ 4 \end{pmatrix}$	6
(2 3)	$\begin{pmatrix} 4 \\ 10 \\ 5 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 2 \\ 5 \\ 1 \end{pmatrix}$	8
(2 4)	$\begin{pmatrix} 4 \\ 1 \\ 10 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 3 \end{pmatrix}$	6

Swaps	New b	$A \otimes \bar{x}$	slack
(3 4)	$\begin{pmatrix} 4 \\ 5 \\ 1 \\ 10 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 5 \\ 1 \\ 7 \end{pmatrix}$	3

Table 8.2: The Second Iteration of the Full Local Search.

From Table 8.2, we can see that the vector $(4, 5, 1, 10)^T$ gives the best slack therefore we will use this vector as the new b and finally we will get:

Swaps	New b	$A \otimes \bar{x}$	slack
(1 2)	$\begin{pmatrix} 5 \\ 4 \\ 1 \\ 10 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 4 \\ 1 \\ 6 \end{pmatrix}$	4
(1 3)	$\begin{pmatrix} 1 \\ 5 \\ 4 \\ 10 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 2 \\ 1 \\ 4 \end{pmatrix}$	11
(1 4)	$\begin{pmatrix} 10 \\ 5 \\ 1 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 4 \\ 4 \\ 6 \end{pmatrix}$	9
(2 3)	$\begin{pmatrix} 4 \\ 1 \\ 5 \\ 10 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 3 \end{pmatrix}$	7
(2 4)	$\begin{pmatrix} 4 \\ 10 \\ 1 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 3 \\ 1 \\ 5 \end{pmatrix}$	7
(3 4)	$\begin{pmatrix} 4 \\ 5 \\ 10 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 4 \\ 3 \\ 6 \\ 1 \end{pmatrix}$	4

Table 8.3: The Third iteration of the Full Local Search.

Therefore the slack cannot be improved anymore and we will conclude the vector $(4, 5, 1, 10)^T$ is the best output found using this method with slack 3. In fact this output is the optimal solution for this problem (this can be checked by considering all $4! = 24$ permutations).

Out of interest we may want to know the output we will get if we use the vector $(1, 5, 4, 10)^T$ instead of the one we have chosen which is $(4, 5, 10, 1)^T$ when there were two choices of vector earlier in the calculation. Now if we use the vector $(1, 5, 4, 10)^T$ as our new b then the result will be the following:

Swaps	New b	$A \otimes \bar{x}$	slack
(1 2)	$\begin{pmatrix} 5 \\ 1 \\ 4 \\ 10 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 3 \end{pmatrix}$	7
(1 3)	$\begin{pmatrix} 4 \\ 5 \\ 1 \\ 10 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 5 \\ 1 \\ 7 \end{pmatrix}$	3
(1 4)	$\begin{pmatrix} 10 \\ 5 \\ 4 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}$	8
(2 3)	$\begin{pmatrix} 1 \\ 4 \\ 5 \\ 10 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 4 \\ 4 \\ 6 \end{pmatrix}$	4
(2 4)	$\begin{pmatrix} 1 \\ 10 \\ 4 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 3 \\ 4 \\ 5 \end{pmatrix}$	7
(3 4)	$\begin{pmatrix} 1 \\ 5 \\ 10 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 2 \\ 4 \\ 4 \end{pmatrix}$	6

Table 8.4: The Second Iteration of the Full Local Search when a different vector is chosen.

As we can see that the vector $(4, 5, 1, 10)^T$ appeared here as well and therefore in this case, we will get the optimal solution regardless of the choice we made earlier on.

8.2.2 Semi-full Local Search

As we can see from the Example 8.2.1, Algorithm 9 will result in $\frac{1}{2}m(m-1)$ different checks if we are looking for all two elements to swap at a time. This means that we will have to check $\frac{1}{2}m(m-1)$ times for every step of the algorithm. If we are looking for more elements to swap at a time, the total check will increase significantly. Therefore we may want to consider another method that would require less computational time to calculate an output in the case when m is very large.

In order to reduce the number of calculations, instead of swapping all $\frac{1}{2}m(m-1)$ pairs of components, we swap one selected component with all $m-1$ remaining swaps. We will use the swap that results in the greatest decrease of slack. This method will only require $m-1$ checks in every step but this will likely to decrease the quality of the output. Therefore the choice of the element we will want to swap first is very important and we would want to choose one such that the quality of the output is reduced as little as possible.

A possible good candidate for the choice could be the entry in b which contributes to the slack, i.e. the k^{th} entry where

$$slk(A,b) = \max_{i=1,\ldots,m} |(A \otimes \bar{x})_i - b_i| = |(A \otimes \bar{x})_k - b_k|.$$

This is the entry in b which is furthest away from the vector $(\bar{b} = A \otimes \bar{x})$ where $\bar{x} = A^* \otimes' b$. The intuitive reason of choosing this entry to swap is because it looks like we will get a better slack if we choose this entry to swap. Also we know that this entry is not hard to find. The following example will illustrate the working for this method.

Example 8.2.2. Given $A = \begin{pmatrix} 1 & 1 & 3 \\ 2 & 4 & 2 \\ 4 & -1 & 5 \\ 3 & 6 & 0 \end{pmatrix}$ and $b = \begin{pmatrix} 10 \\ 5 \\ 4 \\ 1 \end{pmatrix}$.
Find a permutation π s.t. $slk(A, b(\pi))$ is minimized.

We will again use the $identity$ as our starting permutation. Therefore from Example 8.2.2 we know that the resulting slack will be 8.

Now this permutation does not give b as an image of A, therefore we would want to find two entries of b to swap; one of the entries we would like to swap would be the first entry of b since we obtained the slack by this entry. We would swap the first entry with the other three to find out if any one of them will give a better slack.

Swaps	New b	$A \otimes \bar{x}$	slack
(1 2)	$\begin{pmatrix} 5 \\ 10 \\ 4 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}$	9
(1 3)	$\begin{pmatrix} 4 \\ 5 \\ 10 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 4 \\ 3 \\ 6 \\ 1 \end{pmatrix}$	4
(1 4)	$\begin{pmatrix} 1 \\ 5 \\ 4 \\ 10 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 4 \\ 4 \\ 6 \end{pmatrix}$	4

Table 8.5: The First Iteration of the Semi-full Local Search.

Again we can see that if we swap the first and the third entry or the first and the fourth entry, we will obtain a better a slack. Therefore we will choose the vector $(4, 5, 10, 1)^T$ as our new b by the same reason as in the previous example. Now we will repeat the process again. From Table 8.5, we can see that the entry that gives the slack is the third entry. So we swap the third entry with other three entries and we will get:

Swaps	New b	$A \otimes \bar{x}$	slack
(3 1)	$\begin{pmatrix} 10 \\ 5 \\ 4 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}$	8
(3 2)	$\begin{pmatrix} 4 \\ 10 \\ 5 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 2 \\ 5 \\ 1 \end{pmatrix}$	8
(3 4)	$\begin{pmatrix} 4 \\ 5 \\ 1 \\ 10 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 5 \\ 1 \\ 7 \end{pmatrix}$	3

Table 8.6: The Second Iteration of the Semi-full Local Search.

So the vector $(4, 5, 1, 10)^T$ will give a better slack than before and we will use this vector as the new b and after all the calculations we will get:

Swaps	New b	$A \otimes \bar{x}$	slack
(4 1)	$\begin{pmatrix} 10 \\ 5 \\ 1 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 2 \\ 1 \\ 4 \end{pmatrix}$	11
(4 2)	$\begin{pmatrix} 4 \\ 10 \\ 1 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 3 \\ 1 \\ 5 \end{pmatrix}$	7
(4 3)	$\begin{pmatrix} 4 \\ 5 \\ 10 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 4 \\ 3 \\ 6 \\ 1 \end{pmatrix}$	4

Table 8.7: The Third Iteration of the Semi-full Local Search.

Therefore the slack cannot be improved anymore and we will conclude that the vector $(4, 5, 1, 10)^T$ is the best output found using this method with slack 3. Note that for this example the method also gives out the optimal solution for this problem and it also uses the same number of steps as Algorithm 10 but with fewer calculations.

Now it may be interesting to know what will happen of we use the other vector which is $(1, 5, 4, 10)^T$ instead of the one we have chosen earlier. Therefore we will use the vector $(1, 5, 4, 10)^T$ as our b and we will obtain the following table:

Swaps	New b	$A \otimes \bar{x}$	slack
(4 1)	$\begin{pmatrix} 10 \\ 5 \\ 4 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}$	8
(4 2)	$\begin{pmatrix} 1 \\ 10 \\ 4 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 3 \\ 4 \\ 5 \end{pmatrix}$	7
(4 3)	$\begin{pmatrix} 1 \\ 5 \\ 10 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 2 \\ 4 \\ 4 \end{pmatrix}$	6

Table 8.8: The Second Iteration of the Semi-full Local Search when a different vector is chosen.

We can clearly see that we cannot improve our output anymore, therefore if we choose the second vector we will conclude that the vector $(1, 5, 4, 10)^T$ with the slack 4 will be the output. Unlike the full local search method we cannot find a better output with this vector. This shows that this method may sacrifice accuracy in return for the decrease in calculation.

Algorithm 10.

Input: A matrix $A \in \mathbb{R}^{m\times n}$, a vector $b \in \mathbb{R}^m$ and a starting permutation π

Output: A permutation $\bar{\pi}$ and $slk(A, b(\bar{\pi}))$

Set $z = 1$ and $\pi_z := \pi$.

Set $b := b(\pi)$ and find $slk(A, b)$.

While $slk(A, b) \neq 0$

  Find r s.t. $b_r - (A \otimes (A^* \otimes b))_r = slk(A, b)$.

  Find s s.t.

$$slk(A, b(r, s)) = \min_{j=1,\dots,m} slk(A, b(r, j)).$$

  If $slk(A, b(r, s)) < slk(A, b)$

    Set $\pi_{z+1} := \pi_z(r, s)$ and $z := z + 1$.

  else

    $slk(A, b)$ cannot be improved anymore. Therefore $\bar{\pi} := \pi_z$ and

    $slk(A, b(\bar{\pi})) := slk(A, b(\pi_z))$. Stop.

$slk(A, b(\pi_z)) = 0$ implies that $b(\pi_z)$ is an image of A and therefore $\bar{\pi} := \pi_z$, $slk(A, b(\bar{\pi}_z)) = 0$.

8.3 The Column Maxima Method

Consider the following equalities:

$$\begin{aligned}
slk(A,b) &= \max_{k=1,\dots,m} |(A \otimes (\bar{x}))_k - b_k| \\
&= \max_{k=1,\dots,m} |(A \otimes (A^* \otimes' b))_k - b_k| \\
&= \max_{k=1,\dots,m} |(\max_{j=1,\dots,n} (a_{ij} + (\min_{i=1,\dots,m} (-a_{ij} + b_i))))_k - b_k| \\
&= \max_{k=1,\dots,m} |(\max_{j=1,\dots,n} (a_{ij} - \max_{i=1,\dots,m} (a_{ij} - b_i)))_k - b_k| \\
&= \max_{k=1,\dots,m} |\max_{j=1,\dots,n} (a_{kj} - \max_{i=1,\dots,m} (a_{ij} - b_i)) - b_k| \\
&= \max_{k=1,\dots,m} |\max_{j=1,\dots,n} ((a_{kj} - b_k) - \max_{i=1,\dots,m} (a_{ij} - b_i))| \\
&= \max_{k=1,\dots,m} |-\min_{j=1,\dots,n} (\max_{i=1,\dots,m} (a_{ij} - b_i) - (a_{kj} - b_k))| \\
&= \max_{k=1,\dots,m} (\min_{j=1,\dots,n} (\max_{i=1,\dots,m} (a_{ij} - b_i) - (a_{kj} - b_k)))
\end{aligned} \tag{8.1}$$

Now we will let the matrix $\bar{A} = \bar{a}_{ij} = (a_{ij} - b_i)$, i.e. we normalize A, then the equation 8.1 will become

$$slk(A,b) = \max_{k=1,\dots,m} (\min_{j=1,\dots,n} (\max_{i=1,\dots,m} (\bar{a}_{ij}) - \bar{a}_{kj}))$$

where

$$\max_{i=1,\dots,m} \bar{a}_{ij}$$

is the column maximum of column j in $\bar{A}$.

We will suppose that row r in A gives the slack $slk(A,b)$, i.e.

$$\begin{aligned}
slk(A,b) &= \min_{j=1,\dots,n} (\max_{i=1,\dots,m} (a_{ij} - b_i) - (a_{rj} - b_r)) \\
&= \min_{j=1,\dots,n} (\max_{i=1,\dots,m} (\bar{a}_{ij}) - \bar{a}_{rj})).
\end{aligned} \tag{8.2}$$

From the above equation, we can see that the slack can be found by subtracting every column maximum of the matrix $\bar{A}$ by the corresponding values on row r and take the minimal from these values.

Suppose that we have used a starting permutation and find out at which row in A, the slack for this permutation is attained. Now we will want to find another permutation such that the slack will improve. The new permutation is obtained by finding two components in b and swap them. The next step will be to find which two components to swap.

Let us look at (8.2) again, we can see that there are two ways to reduce the slack:

1) We will look for an entry b_s to swap with the value for b_r such that every entry of row r in $\bar{A}$ will increase, i.e. $\forall j \in N$,

$$a_{rj} - b_s > a_{rj} - b_r = \bar{a}_{rj}.$$

Then the difference between the values of row r and the column maxima are decreased after the swap and the slack will decrease as the result.

2) We find the column l such that the minimal value between the column maximum and the value on row r is attained, i.e.

$$slk(A, b) = \max_{i=1,\ldots,m} (\bar{a}_{il} - \bar{a}_{rl})$$

Then we will find an entry b_q to swap with the value b_p where $\bar{a}_{pl}$ is the column maximum of column l such that

$$a_{pl} - b_q < a_{pl} - b_p = \bar{a}_{pl}.$$

Therefore the column maximum for column l is decreased after the swap and hence the slack will decrease.

8.3.1 Formulation of the Algorithm

Step 1. The first step of this method will require us to have a starting permutation, the identity for instance and we will use this permutation to find out $\bar{A}$. Using $\bar{A}$ we can find out the slack for this permutation and also the row r in which this slack is obtained. We will also want to acquire the set $\bar{M} \subseteq M$ where $\bar{M}$ is the set of indices which are covered by a column maximum in $\bar{A}$, i.e.

$$\bar{M} = \{k \in M \mid \exists j \in N,\ \bar{a}_{kj} = \max_{i \in M} \bar{a}_{ij}\}$$

or equivalently

$$\bar{M} = \bigcup_{j \in N} M_j(A, b)$$

where

$$M_j(A, b) = \{k \in M \mid (a_{kj} - b_k) = \max_{i=1,\dots,m} (a_{ij} - b_i)\}$$

which were seen previously in different chapters.

Note that if there are more than one row in which the slack is obtained then we will choose the row with the largest value of b as b_r.

Example 8.3.1.

$$\begin{pmatrix} \qquad\qquad \\ \\ \hline \\ \hline \\ \hline \\ \end{pmatrix} \begin{array}{c} \\ \\ \\ \text{row } u \\ \text{row } v \\ \\ \end{array} \begin{pmatrix} \cdot \\ \cdot \\ b_u \\ b_v \\ \cdot \end{pmatrix}$$

From the matrix above, the rows u and v represent the rows in which the slack is obtained. Therefore we will choose the row which has the largest value for b, therefore we have $b_r =$

$\max(b_u, b_v)$.

Step 2. The next step will be to find out which component in b we will need to swap from. We will first look at row r and try to increase every value of this row in $\bar{A}$ so that the slack is reduced. In order to do this we will need to find b_s such that $\forall j \in N$,

$$a_{rj} - b_s > a_{rj} - b_r$$

or

$$b_s < b_r.$$

We would also like to choose b_s where there exists $j \in N$ such that $\bar{a}_{sj}$ is a column maximum in $\bar{A}$, i.e.

$$s \in M_j(A, b)$$

or equivalently,

$$\bar{a}_{sj} = \max_{i \in M} \bar{a}_{ij}.$$

This is because we would want to reduce the value of the column maximum at the same time. We also want to decrease the slack in small steps to avoid the slack going past a local minimum. Therefore if there exists more than one choice for b_s, we would like to choose the largest from them all, i.e.

$$b_r - b_s = \min_{i \in \bar{M}}((b_r - b_i) > 0).$$

We immediately see that b_s is the smaller closest value of b_r. After we found this b_s we will perform the swap and obtain the new $\bar{A}$. We will calculate the resulting slack from the new $\bar{A}$. If the value is better than we obtained previously, then we will repeat this process with the new $\bar{A}$ from **Step 1**.

Step 3. If we cannot find a value of b_s which satisfies the criteria above or we have obtained a worse slack from the previous step, we will move on to this step. We will need to obtain all the column maxima such that the minimal value between these column maxima and the value on row r is attained. In the case when there are more than one row with a column maximum which contribute to the slack, we choose the row p such that b_p has the smallest value of b from the other row.

Example 8.3.2.

$$\left(\begin{array}{ccc} & & \\ & \bigcirc & \\ \bigcirc & \bigcirc & \\ & & \bigcirc \\ & & \\ \hline & & \\ \hline & & \end{array}\right) \begin{array}{c} \\ \text{row } u \\ \text{row } v \\ \text{row } w \\ \\ \text{row } r \\ \\ \end{array} \begin{pmatrix} . \\ b_u \\ b_v \\ b_w \\ . \\ b_r \\ . \end{pmatrix} \qquad \begin{array}{c} \bigcirc \text{ are the column maxima} \\ \text{which contribute} \\ \text{to the slack} \end{array}$$

From the matrix above $\bigcirc$ represents the column maximum which contribute to the slack, then we would have row u, v and w to choose as from. Now we will choose the row which has the smallest value for b, hence we have $b_p = \min(b_u, b_v, b_w)$.

Now we want to decrease the column maxima of row p by choosing a value b_q such that

$$a_{pl} - b_q < a_{pl} - b_p$$

or

$$b_q > b_p.$$

We also want to choose b_q such that row q in $\bar{A}$ do not have a column maximum, i.e. $\forall j \in N$,

$$q \notin M_j(A, b)$$

or equivalently,

$$\bar{a}_{qj} \neq \max_{i \in M} \bar{a}_{ij}.$$

By the similar reason as **Step 2**, if there exists more than one choice for b_q, we would want to choose the smallest of them all. This is because we want the slack to decrease in small steps, i.e.

$$b_p - b_q = \max_{i \notin \bar{M}}((b_p - b_i) < 0).$$

We will perform the swap and obtain the resulting slack from the new $\bar{A}$. If the value is better than that we obtained previously, we will start from **Step 1** again using the new permutation. If we cannot find a b_q for the swap or if we obtain a worse output after the swap, we will say that the output cannot improve. A local minimum is obtained by the permutation we used before the swap and this is the end of algorithm.

There are a couple of points we will need to consider during the run of this algorithm:

- If we have obtained a permutation π such that $|\bar{M}| = m$, this means that there exists at least one column maximum in every row of $\bar{A}$. This immediately implies that $b(\pi)$ is an image of A. Hence $slk(A, b(\pi)) = 0$ and we can stop the algorithm with this permutation.

- If we have a repeated permutation after we applied **Step 2** or **Step 3**, then we will have a cycle and therefore we cannot find a better output after this point. We will stop the algorithm here and we will take this permutation to be the output of this algorithm.

Algorithm 11.

Input: A matrix $A \in \mathbb{R}^{m \times n}$, a vector $b \in \mathbb{R}^m$ and a starting permutation π

Output: A permutation $\bar{\pi}$ and $slk(A, b(\bar{\pi}))$

Set $z = 1$, $\pi_z := \pi$, $b := b(\pi)$ and $\bar{A} := \bar{a}_{ij} = (a_{ij} - b_i)$.

Find $slk(A, b)$ and

$$\bar{M} := \{k \in M \mid \exists j \in N,\ \bar{a}_{kj} = \max_{i \in M} \bar{a}_{ij}\}.$$

While $|\bar{M}| \neq m$

If $\pi_z \neq \pi_i, \ \ i = 1, ..., (z-1)$

Find the set

$$R := \{l \notin \bar{M} \mid \min_{j \in N}(\max_{i \in M}(\bar{a}_{ij}) - \bar{a}_{lj})) = slk(A, b)\}.$$

Find r s.t.

$$b_r = \max_{i \in R} b_i$$

If $\exists s$ s.t.

$$b_r - b_s = \min_{i \in \bar{M}}((b_r - b_i) > 0)$$

and $slk(A, b(r, s)) \leq slk(A, b)$.

Set $b := b(r, s)$, $\pi_{z+1} := (r, s)\pi_z$, $z := z + 1$ and find the new $\bar{M}$

Else

Find the set

$$P := \{u \in M \mid \min_{j \in N}(\bar{a}_{uj} - \bar{a}_{rj}) = slk(A, b) \text{ and}$$

$$\bar{a}_{uj} = \max_{i \in M} \bar{a}_{ij}\}.$$

Find p s.t.

$$b_p = \min_{i \in P} b_i$$

If $\exists q$ s.t.

$$b_p - b_q = \max_{i \notin \bar{M}}((b_p - b_i) < 0)$$

and $slk(A, b(p,q)) \leq slk(A, b)$.

Set $b := b(p,q)$, $\pi_{z+1} := (p,q)\pi_z$, $z := z + 1$ and find the new $\bar{M}$

Else

$slk(A, b)$ cannot be improved anymore. Therefore $\bar{\pi} := \pi_z$ and $slk(A, b(\bar{\pi})) := slk(A, b(\pi_z))$. Stop.

Stop.

Stop.

else

π_z is a repeated permutation and we will not obtain a better solution beyond this point. Therefore $\bar{\pi} := \pi_z$ and $slk(A, b(\bar{\pi})) := slk(A, b(\pi_z))$. Stop.

Stop.

else

$|\bar{M}| = m$ implies that $b(\pi_z)$ is an image of A and hence $slk(A, b(\pi_z)) = 0$. Therefore $\bar{\pi} := \pi_z$ and $slk(A, b(\bar{\pi}_z)) = 0$. Stop.

8.4 Test Results for the Three Methods

In this section we will run tests on the three algorithms arising from the three methods and compare the results obtained. We will run tests on 20 different sets of problem, i.e. we have 20 distinct sets of matrix A and vector b. For each set of problem we will use 7 different permutations as our starting permutations; the identity, Atimes0 and five other randomly generated permutations. The permutation Atimes0 is generated by considering the vector $A \otimes 0$ as a one column problem and Atimes0 will be a solution to this problem.

In order to assess the quality of the methods, the first four sets of problem are randomly generated which consists of only two columns. Using Algorithm 8, we can solve these problems precisely and therefore the best slack is known for all these four cases.

For the rest of the sets, we have generated a random matrix A with specified number of rows and columns with b is generated by choosing randomly permuted of a random image of A. Therefore we know that the minimum slack will be 0 for all these cases. The results will be presented in the following pages.

The following table represents results obtained by using Full Local Search Method:

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
1	10	2	2131	Identity	7309	2	$< 1s$
				Atimes0	6824	3	$< 1s$
				Random 1	7011	3	$< 1s$
				Random 2	5653	2	$< 1s$
				Random 3	4770	2	$< 1s$
				Random 4	3964	5	$< 1s$
				Random 5	6341	2	$< 1s$
2	50	2	2343	Identity	13806	2	$< 1s$
				Atimes0	12240	3	$< 1s$
				Random 1	12462	3	$< 1s$
				Random 2	14720	2	$< 1s$
				Random 3	14015	1	$< 1s$
				Random 4	11185	2	$< 1s$
				Random 5	12828	2	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
3	100	2	1503	Identity	12871	2	$10s$
				Atimes0	14344	2	$10s$
				Random 1	13714	3	$14s$
				Random 2	11124	3	$14s$
				Random 3	15622	1	$5s$
				Random 4	12803	5	$23s$
				Random 5	14579	1	$5s$
4	200	2	892	Identity	16702	2	$1m37s$
				Atimes0	16059	4	$3m17s$
				Random 1	15127	2	$1m41s$
				Random 2	15026	3	$2m30s$
				Random 3	15127	2	$1m41s$
				Random 4	14856	1	$49s$
				Random 5	16095	2	$1m39s$
5	10	10	0	Identity	583	2	$< 1s$
				Atimes0	466	2	$< 1s$
				Random 1	321	3	$< 1s$
				Random 2	726	2	$< 1s$
				Random 3	841	2	$< 1s$
				Random 4	827	3	$< 1s$
				Random 5	440	3	$< 1s$
6	50	10	0	Identity	3023	2	$2s$
				Atimes0	2628	2	$2s$
				Random 1	2787	2	$2s$
				Random 2	2868	3	$3s$
				Random 3	2976	1	$1s$
				Random 4	2692	2	$2s$
				Random 5	2642	3	$3s$
7	100	10	0	Identity	3979	2	$25s$
				Atimes0	4271	4	$53s$
				Random 1	3898	3	$38s$
				Random 2	4699	2	$26s$
				Random 3	4462	2	$26s$
				Random 4	4389	2	$26s$
				Random 5	4715	2	$25s$
8	200	10	0	Identity	5321	2	$5m41s$
				Atimes0	5050	2	$5m42s$
				Random 1	4810	2	$5m47s$
				Random 2	4851	3	$8m39s$
				Random 3	4786	2	$5m43s$
				Random 4	4748	3	$8m35s$
				Random 5	4586	4	$11m26s$
9	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
10	50	50	0	Identity	529	2	$7s$
				Atimes0	427	2	$7s$
				Random 1	454	3	$11s$
				Random 2	390	3	$11s$
				Random 3	416	3	$11s$
				Random 4	507	2	$7s$
				Random 5	382	4	$15s$
11	100	50	0	Identity	1185	2	$1m47s$
				Atimes0	916	3	$2m54s$
				Random 1	994	2	$1m57s$
				Random 2	933	4	$4m$
				Random 3	1049	3	$2m57s$
				Random 4	1002	2	$1m50s$
				Random 5	1114	2	$1m47s$
12	200	50	0	Identity	1159	2	$26m9s$
				Atimes0	1096	4	$50m29s$
				Random 1	1056	4	$50m36s$
				Random 2	1236	2	$25m22s$
				Random 3	1228	2	$25m19s$
				Random 4	1237	3	$37m54a$
				Random 5	1126	4	$50m27s$
13	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
14	50	100	0	Identity	214	2	$14s$
				Atimes0	218	2	$14s$
				Random 1	271	2	$14s$
				Random 2	173	3	$21s$
				Random 3	103	2	$14s$
				Random 4	210	3	$21s$
				Random 5	124	2	$14s$
15	100	100	0	Identity	537	2	$3m18s$
				Atimes0	419	2	$3m27s$
				Random 1	408	3	$5m9s$
				Random 2	527	2	$3m26s$
				Random 3	491	2	$3m25s$
				Random 4	520	2	$3m28s$
				Random 5	393	3	$5m4s$
16	200	100	0	Identity	701	2	52m4s
				Atimes0	737	3	$1h16m55s$
				Random 1	681	3	$1h14m47s$
				Random 2	735	1	$24m56s$
				Random 3	713	3	$1h14m42s$
				Random 4	704	4	$1h42m14s$
				Random 5	716	3	$1h16m46s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
17	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
18	50	200	0	Identity	0	1	$14s$
				Atimes0	56	1	$13s$
				Random 1	2	2	$27s$
				Random 2	29	2	$27s$
				Random 3	0	3	$40s$
				Random 4	0	1	$13s$
				Random 5	0	1	$13s$
19	100	200	0	Identity	293	2	$6m39s$
				Atimes0	212	2	$6m34s$
				Random 1	244	3	$9m45s$
				Random 2	208	5	$16m14s$
				Random 3	194	3	$9m45s$
				Random 4	263	2	$6m30s$
				Random 5	236	2	$6m31s$
20	200	200	0	Identity	408	2	$1h38m32s$
				Atimes0	372	3	$2h33m16s$
				Random 1	364	3	$2h33m40s$
				Random 2	327	2	$1h44m32s$
				Random 3	399	2	$1h42m17s$
				Random 4	391	2	$1h38m42s$
				Random 5	352	4	$3h22m37s$

Table 8.9: Results obtained using Full Local Search Method for 20 matrices with different dimensions.

The following table represents results obtained by using Semi-full Local Search Method:

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
1	10	2	2131	Identity	7309	2	$< 1s$
				Atimes0	8018	2	$< 1s$
				Random 1	7498	3	$< 1s$
				Random 2	7019	3	$< 1s$
				Random 3	10122	2	$< 1s$
				Random 4	8317	3	$< 1s$
				Random 5	8134	3	$< 1s$
2	50	2	2343	Identity	13806	2	$< 1s$
				Atimes0	14410	2	$< 1s$
				Random 1	16160	1	$< 1s$
				Random 2	14720	2	$< 1s$
				Random 3	14015	1	$< 1s$
				Random 4	12648	3	$< 1s$
				Random 5	14141	2	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
3	100	2	1503	Identity	12871	2	$< 1s$
				Atimes0	14798	2	$< 1s$
				Random 1	15491	1	$< 1s$
				Random 2	15053	3	$< 1s$
				Random 3	15622	1	$< 1s$
				Random 4	15935	3	$< 1s$
				Random 5	14579	1	$< 1s$
4	200	2	892	Identity	16702	2	$< 1s$
				Atimes0	16973	1	$< 1s$
				Random 1	15280	2	$< 1s$
				Random 2	16005	3	$1s$
				Random 3	15932	2	$< 1s$
				Random 4	14856	1	$< 1s$
				Random 5	16369	2	$< 1s$
5	10	10	0	Identity	868	2	$< 1s$
				Atimes0	1012	2	$< 1s$
				Random 1	708	3	$< 1s$
				Random 2	1055	2	$< 1s$
				Random 3	1846	1	$< 1s$
				Random 4	1190	1	$< 1s$
				Random 5	587	3	$< 1s$
6	50	10	0	Identity	3052	2	$< 1s$
				Atimes0	2768	3	$< 1s$
				Random 1	2827	4	$< 1s$
				Random 2	2936	3	$< 1s$
				Random 3	2976	1	$< 1s$
				Random 4	2692	2	$< 1s$
				Random 5	2809	2	$< 1s$
7	100	10	0	Identity	3979	2	$< 1s$
				Atimes0	4598	3	$< 1s$
				Random 1	4590	3	$< 1s$
				Random 2	4749	2	$< 1s$
				Random 3	4706	2	$< 1s$
				Random 4	4487	2	$< 1s$
				Random 5	4724	2	$< 1s$
8	200	10	0	Identity	5364	2	$3s$
				Atimes0	5050	2	$3s$
				Random 1	4965	2	$3s$
				Random 2	5185	1	$2s$
				Random 3	4786	2	$3s$
				Random 4	5121	3	$5s$
				Random 5	4888	4	$7s$
9	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
10	50	50	0	Identity	529	2	$< 1s$
				Atimes0	597	1	$< 1s$
				Random 1	611	1	$< 1s$
				Random 2	463	2	$< 1s$
				Random 3	606	2	$< 1s$
				Random 4	675	1	$< 1s$
				Random 5	526	1	$< 1s$
11	100	50	0	Identity	1185	2	$2s$
				Atimes0	895	4	$5s$
				Random 1	984	3	$4s$
				Random 2	1126	2	$2s$
				Random 3	1049	3	$4s$
				Random 4	1002	2	$2s$
				Random 5	1219	1	$1s$
12	200	50	0	Identity	1159	2	$15s$
				Atimes0	1341	1	$8s$
				Random 1	1096	3	$23s$
				Random 2	1275	2	$16s$
				Random 3	1195	3	$23s$
				Random 4	1237	3	$23s$
				Random 5	1206	2	$16s$
13	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
14	50	100	0	Identity	214	2	$< 1s$
				Atimes0	203	4	$1s$
				Random 1	453	1	$< 1s$
				Random 2	383	1	$< 1s$
				Random 3	217	2	$< 1s$
				Random 4	237	2	$< 1s$
				Random 5	124	2	$< 1s$
15	100	100	0	Identity	574	2	$4s$
				Atimes0	419	2	$4s$
				Random 1	408	3	$6s$
				Random 2	639	2	$4s$
				Random 3	581	2	$4s$
				Random 4	578	2	$4s$
				Random 5	470	3	$6s$
16	200	100	0	Identity	731	2	$30s$
				Atimes0	812	1	$15s$
				Random 1	704	3	$45s$
				Random 2	735	1	$15s$
				Random 3	829	1	$15s$
				Random 4	843	2	$30s$
				Random 5	716	3	$45s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
17	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
18	50	200	0	Identity	12	2	$1s$
				Atimes0	56	1	$< 1s$
				Random 1	55	3	$2s$
				Random 2	63	2	$1s$
				Random 3	127	1	$< 1s$
				Random 4	0	1	$< 1s$
				Random 5	0	1	$< 1s$
19	100	200	0	Identity	293	2	$8s$
				Atimes0	212	2	$8s$
				Random 1	320	2	$8s$
				Random 2	292	2	$8s$
				Random 3	251	2	$8s$
				Random 4	275	2	$8s$
				Random 5	274	1	$4s$
20	200	200	0	Identity	408	2	$1m$
				Atimes0	386	3	$1m29s$
				Random 1	369	3	$1m33s$
				Random 2	353	2	$59s$
				Random 3	409	2	$59s$
				Random 4	400	4	$1m59s$
				Random 5	368	4	$1m59s$

Table 8.10: Results obtained using Semi-full Local Search Method for 20 matrices with different dimensions.

The following table represents results obtained by using the Column Maxima Method:

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
1	10	2	2131	Identity	9979	1	$< 1s$
				Atimes0	8275	3	$< 1s$
				Random 1	10238	2	$< 1s$
				Random 2	9999	1	$< 1s$
				Random 3	8276	3	$< 1s$
				Random 4	10371	2	$< 1s$
				Random 5	9938	4	$< 1s$
2	50	2	2343	Identity	13059	7	$< 1s$
				Atimes0	13579	5	$< 1s$
				Random 1	10242	4	$< 1s$
				Random 2	11376	6	$< 1s$
				Random 3	12726	3	$< 1s$
				Random 4	13976	6	$< 1s$
				Random 5	12457	6	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
3	100	2	1503	Identity	12425	6	$< 1s$
				Atimes0	13047	7	$< 1s$
				Random 1	12608	13	$< 1s$
				Random 2	15858	5	$< 1s$
				Random 3	12969	6	$< 1s$
				Random 4	14793	5	$< 1s$
				Random 5	13029	6	$< 1s$
4	200	2	892	Identity	14727	10	$< 1s$
				Atimes0	14692	5	$< 1s$
				Random 1	14337	8	$< 1s$
				Random 2	16196	3	$< 1s$
				Random 3	16751	6	$< 1s$
				Random 4	12910	14	$< 1s$
				Random 5	13717	11	$< 1s$
5	10	10	0	Identity	1362	0	$< 1s$
				Atimes0	1572	0	$< 1s$
				Random 1	457	1	$< 1s$
				Random 2	877	4	$< 1s$
				Random 3	1572	1	$< 1s$
				Random 4	1172	1	$< 1s$
				Random 5	2669	2	$< 1s$
6	50	10	0	Identity	3095	2	$< 1s$
				Atimes0	3148	2	$< 1s$
				Random 1	4379	1	$< 1s$
				Random 2	3574	1	$< 1s$
				Random 3	2832	1	$< 1s$
				Random 4	2952	2	$< 1s$
				Random 5	3308	1	$< 1s$
7	100	10	0	Identity	4456	6	$< 1s$
				Atimes0	4305	2	$< 1s$
				Random 1	5182	3	$< 1s$
				Random 2	3833	7	$< 1s$
				Random 3	5654	0	$< 1s$
				Random 4	4698	0	$< 1s$
				Random 5	4404	5	$< 1s$
8	200	10	0	Identity	5948	0	$< 1s$
				Atimes0	5085	5	$< 1s$
				Random 1	4487	5	$< 1s$
				Random 2	4740	3	$< 1s$
				Random 3	5300	3	$< 1s$
				Random 4	4814	7	$< 1s$
				Random 5	5596	1	$< 1s$
9	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
10	50	50	0	Identity	557	2	$< 1s$
				Atimes0	597	0	$< 1s$
				Random 1	308	8	$< 1s$
				Random 2	643	1	$< 1s$
				Random 3	502	5	$< 1s$
				Random 4	509	1	$< 1s$
				Random 5	333	3	$< 1s$
11	100	50	0	Identity	1136	3	$< 1s$
				Atimes0	1297	3	$< 1s$
				Random 1	782	8	$2s$
				Random 2	1100	4	$< 1s$
				Random 3	1085	6	$< 1s$
				Random 4	1014	6	$< 1s$
				Random 5	1066	3	$< 1s$
12	200	50	0	Identity	1187	3	$1s$
				Atimes0	1072	4	$2s$
				Random 1	1263	3	$2s$
				Random 2	1361	3	$1s$
				Random 3	1290	4	$2s$
				Random 4	1015	12	$5s$
				Random 5	1438	4	$2s$
13	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
14	50	100	0	Identity	190	5	$< 1s$
				Atimes0	482	0	$< 1s$
				Random 1	118	9	$1s$
				Random 2	216	3	$< 1s$
				Random 3	146	12	$2s$
				Random 4	192	3	$< 1s$
				Random 5	510	0	$< 1s$
15	100	100	0	Identity	574	2	$1s$
				Atimes0	434	5	$2s$
				Random 1	323	10	$2s$
				Random 2	636	1	$< 1s$
				Random 3	449	4	$1s$
				Random 4	612	3	$2s$
				Random 5	437	6	$2s$
16	200	100	0	Identity	731	2	$3s$
				Atimes0	791	2	$3s$
				Random 1	732	4	$4s$
				Random 2	653	5	$6s$
				Random 3	695	8	$7s$
				Random 4	689	3	$4s$
				Random 5	688	5	$5s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
17	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
18	50	200	0	Identity	0	2	$< 1s$
				Atimes0	56	0	$< 1s$
				Random 1	0	8	$2s$
				Random 2	0	4	$1s$
				Random 3	2	4	$2s$
				Random 4	31	1	$1s$
				Random 5	0	1	$< 1s$
19	100	200	0	Identity	439	0	$2s$
				Atimes0	298	0	$2s$
				Random 1	361	0	$2s$
				Random 2	197	8	$4s$
				Random 3	338	4	$4s$
				Random 4	290	1	$3s$
				Random 5	228	4	$4s$
20	200	200	0	Identity	332	12	$23s$
				Atimes0	476	1	$5s$
				Random 1	399	3	$9s$
				Random 2	338	2	$7s$
				Random 3	403	2	$7s$
				Random 4	338	10	$18s$
				Random 5	332	11	$25s$

Table 8.11: Results obtained using The Column Maxima Method for 20 matrices with different dimensions.

The following table compares the results obtained from the three methods:

Test matrix	m	n	Optimal	Starting permutation	FLS	S-FLS	TCMM
1	10	2	2131	Identity	7309	7309	9979
				Atimes0	6824	8018	8275
				Random 1	7011	7498	10238
				Random 2	5653	7019	9999
				Random 3	4770	10122	8276
				Random 4	3964	8317	10371
				Random 5	6341	8134	9938
2	50	2	2343	Identity	13806	13806	13059
				Atimes0	12240	14410	13579
				Random 1	12462	16160	10242
				Random 2	14720	14720	11376
				Random 3	14015	14015	12726
				Random 4	11185	12648	13976
				Random 5	12828	14141	12457

Test matrix	m	n	Optimal	Starting permutation	FLS	S-FLS	TCMM
3	100	2	1503	Identity	12871	12871	12425
				Atimes0	14344	14798	13047
				Random 1	13714	15491	12608
				Random 2	11124	15053	15858
				Random 3	15622	15622	12969
				Random 4	12803	15935	14793
				Random 5	14579	14579	13029
4	200	2	892	Identity	16702	16702	14727
				Atimes0	16059	16973	14692
				Random 1	15127	15280	14337
				Random 2	15026	16005	16196
				Random 3	15127	15932	16751
				Random 4	14856	14856	12910
				Random 5	16095	16369	13717
5	10	10	0	Identity	583	868	1362
				Atimes0	466	1012	1572
				Random 1	321	708	457
				Random 2	726	1055	877
				Random 3	841	1846	1572
				Random 4	827	1190	1172
				Random 5	440	587	2669
6	50	10	0	Identity	3023	3052	3095
				Atimes0	2628	2768	3148
				Random 1	2787	2827	4379
				Random 2	2868	2936	3574
				Random 3	2976	2976	2832
				Random 4	2692	2692	2952
				Random 5	2642	2809	3308
7	100	10	0	Identity	3979	3979	4456
				Atimes0	4271	4598	4305
				Random 1	3898	4590	5182
				Random 2	4699	4749	3833
				Random 3	4462	4706	5654
				Random 4	4389	4487	4698
				Random 5	4715	4724	4404
8	200	10	0	Identity	5321	5364	5948
				Atimes0	5050	5050	5085
				Random 1	4810	4965	4487
				Random 2	4851	5185	4740
				Random 3	4786	4786	5300
				Random 4	4748	5121	4814
				Random 5	4586	4888	5596
9	10	50	0	Identity	0	0	0
				Atimes0	0	0	0
				Random 1	0	0	0
				Random 2	0	0	0
				Random 3	0	0	0
				Random 4	0	0	0
				Random 5	0	0	0

Test matrix	m	n	Optimal	Starting permutation	FLS	S-FLS	TCMM
10	50	50	0	Identity	529	529	557
				Atimes0	427	597	597
				Random 1	454	611	308
				Random 2	390	463	643
				Random 3	416	606	502
				Random 4	507	675	509
				Random 5	382	526	333
11	100	50	0	Identity	1185	1185	1136
				Atimes0	916	895	1297
				Random 1	994	984	782
				Random 2	933	1126	1100
				Random 3	1049	1049	1085
				Random 4	1002	1002	1014
				Random 5	1114	1219	1066
12	200	50	0	Identity	1159	1159	1187
				Atimes0	1096	1341	1072
				Random 1	1056	096	1263
				Random 2	1236	1275	1361
				Random 3	1228	1195	1290
				Random 4	1237	1237	1015
				Random 5	1126	1206	1438
13	10	50	0	Identity	0	0	0
				Atimes0	0	0	0
				Random 1	0	0	0
				Random 2	0	0	0
				Random 3	0	0	0
				Random 4	0	0	0
				Random 5	0	0	0
14	50	100	0	Identity	214	214	190
				Atimes0	218	203	482
				Random 1	271	453	118
				Random 2	173	383	216
				Random 3	103	217	146
				Random 4	210	237	192
				Random 5	124	124	510
15	100	100	0	Identity	537	574	574
				Atimes0	419	419	434
				Random 1	408	408	323
				Random 2	527	639	636
				Random 3	491	581	449
				Random 4	520	578	612
				Random 5	393	470	437
16	200	100	0	Identity	701	731	731
				Atimes0	737	812	791
				Random 1	681	704	732
				Random 2	735	735	653
				Random 3	713	829	695
				Random 4	704	843	689
				Random 5	716	716	688

Test matrix	m	n	Optimal	Starting permutation	FLS	S-FLS	TCMM
17	10	50	0	Identity	0	0	0
				Atimes0	0	0	0
				Random 1	0	0	0
				Random 2	0	0	0
				Random 3	0	0	0
				Random 4	0	0	0
				Random 5	0	0	0
18	50	200	0	Identity	0	12	0
				Atimes0	56	56	56
				Random 1	2	55	0
				Random 2	29	63	0
				Random 3	0	127	2
				Random 4	0	0	31
				Random 5	0	0	0
19	100	200	0	Identity	293	293	439
				Atimes0	212	212	298
				Random 1	244	320	361
				Random 2	208	292	197
				Random 3	194	251	338
				Random 4	263	275	290
				Random 5	236	274	228
20	200	200	0	Identity	408	408	332
				Atimes0	372	386	476
				Random 1	364	369	399
				Random 2	327	353	338
				Random 3	399	409	403
				Random 4	391	400	338
				Random 5	352	368	332

Table 8.12: Comparison of the results obtained from the three methods.

From Table 8.9, 8.10 and 8.11, we can clearly see that the full local search (FLS) method uses the most time out of the three methods. The semi-full local search (S-FLS) uses considerably less time than FLS which is expected. And we can see that the column maxima method (CMM) is the fastest algorithm out of the three.

When we compare the results obtained from the three algorithms using Table 8.15, we can see that FLS method generally gives a better result than S-FLS. This is also expected due to the fact that we have decreased the quality of the output by reducing the number of calculations. We can see that CMM gives mix results but in general, it did not give a better result than FLS or S-FLS. We can also see that if we choose any one of these seven permutations as a starting permutation, generally we do not obtain a better output than by choosing the other six permutations. This may imply that the choice of starting permutation does not affect the output.

Now we will compare the results with the optimal solutions. For the first four cases, the outputs we have obtained are significantly larger then the optimal solutions. This implies that the three methods did not give a good approximation to the optimal solution. This remains generally the case for most of the other test matrices. We should note that for any fix value of n, the quality of the outputs from the three methods started to decrease when we increased m. In general we have noticed that when m is significantly larger than n we are unable to get a good quality output from the three methods.

We should also pay some attention to the Test matrix 9, 13 and 17 from Table 8.9, 8.10 and 8.11 where $n >> m$. We can see that for these three sets of problem we have obtained the optimal solution by using the starting permutations. This implies that after applying these seven starting permutations to b, the vectors we obtained are still images of A. This could imply that this vector b may remain to be an image of A after any permutations. This may be a significant result in permuted linear system problem and we may wish to investigate this in the future.

We may also try to investigate and develop some other heuristic methods to obtain a better approximate for the permuted linear system problem especially when $m >> n$. There are several promising methods available which include simulated annealing, tabu search, genetic algorithms and particle swarm optimization.

Unfortunately we are unable to obtain a good approximate of the optimal solution from FLS, S-FLS and CMM. But if we look at Table 8.9, 8.10 and 8.11 again we can see that all three algorithms stop after a small number of steps. This implies that the local minima are very close together therefore we would want to modify our methods so that we can move away from a local minimum when we reached it.

One of the methods we can use is simulated annealing and we will combine simulated annealing into the FLS and S-FLS to produce two other heuristic methods.

8.5 Simulated Annealing

Simulated annealing [50] [65] is another local search method in which we use an initial solution as our current solution and then look for a possible solution in the neighbourhood of the current solution. Then we compare the objective function values between the two solutions. If the possible solution gives a better value than the current solution, then we will accept it as our new solution. But in the case when the value from the possible solution is worse than the value given by the current solution, we will find the probability of it to be accepted as our new solution. If the probability of being accepted is high enough then we will use it as our new solution or we will reject it otherwise.

The probability of accepting a worse solution depends on the difference in objective function values between the two solution. If the objective value from the possible solution is a lot worse than the value given by the current solution then it is more likely to be rejected and there is more chance to accept the solution if the difference is small.

This probability also depends on a parameter T in which T is called the *temperature*. The value of T decreases after every step by a ratio which is strictly between 0 and 1. As the temperature decreases the probability of a worse solution being accepted also decreases and we will stop when the temperature is reduced to a certain level, i.e. *frozen*. The following is an outline of the simulated annealing algorithm:

Algorithm 12. (Simulated Annealing algorithm)

Input: Initial solution x, initial temperature T, frozen temperature t and reduction ratio $0 < r < 1$.

Output: Solution $\bar{x}$

While $T \nleq t$

Find a new solution x' in the neighbourhood of x

If $random(0, 1) < \exp \frac{Obj(x)-Obj(x')}{T}$

Set $x := x', T := rT$.

Else

Break.

Algorithm reach ending condition and solution is given by $\bar{x} = x$.

8.5.1 Simulated Annealing Full Local Search

From the test we have done on the algorithm of full local search, semi-full local search and the column maxima method, we can see that alone they do not give us an output close to an optimal solution. Now we apply the method of simulated annealing into the full local search method.

The SAFLS method (short for *Simulated Annealing Full Local Search*) will use the basic structure of Algorithm 12 but we will have permutation as our solution and slacks as our objective function value. Also we will now use the best permutation found by the local search method as our new solution x'. Therefore the SA-FLS algorithm will be the following:

Algorithm 13.

Input: A matrix $A \in \mathbb{R}^{m\times n}$, a vector $b \in \mathbb{R}^m$, a starting permutation π, initial temperature T, frozen temperature t and reduction ratio $0 < r < 1$.

Output: A permutation $\bar{\pi}$ and $slk(A, b(\bar{\pi}))$

Set $z = 1$ and $\pi_z := \pi$.

Set $b := b(\pi)$ and find $slk(A, b)$.

While $T \nleq t$

While $slk(A, b) \neq 0$

For $i = 1$ to m

For $j = 1$ to m, $j \neq i$

Find $slk(A, b(i,j))$.

Let

$$slk(A, b(r,s)) = \min_{i,j}(slk(A, b(i,j))).$$

If

$random(0,1) < \exp \frac{slk(A,b) - slk(A,b(r,s))}{T}$

Set $\pi_{z+1} := \pi_z(r,s)$, $z := z+1$ and $T := rT$.

Else

$slk(A,b)$ cannot be improved anymore. Therefore

$$\bar{\pi} := \min_z(\pi_z)$$

and $slk(A, b(\bar{\pi}))$ is the resulting slack. Stop.

$slk(A, b(\pi_z)) = 0$ implies that $b(\pi_z)$ is an image of A and therefore $\bar{\pi} := \pi_z$ and $slk(A, b(\bar{\pi_z})) = 0$.

Termination criterion is satisfied therefore

$$\bar{\pi} := \min_z(\pi_z)$$

and $slk(A, b(\bar{\pi}))$ is the resulting slack.

8.5.2 Simulated Annealing Semi-Full Local Search

Similarly to the local search method we can also apply the method of simulated annealing into the semi-full local search method and we will have created the SASFLS method (short for *Simulated Annealing Semi-Full Local Search*). The resulting algorithm will be as follows:

Algorithm 14.

Input: A matrix $A \in \mathbb{R}^{m\times n}$, a vector $b \in \mathbb{R}^m$, a starting permutation π, initial temperature T, frozen temperature t and reduction ratio $0 < r < 1$.

Output: A permutation $\bar{\pi}$ and $slk(A, b(\bar{\pi}))$

Set $z = 1$ and $\pi_z := \pi$.

Set $b := b(\pi)$ and find $slk(A, b)$.

While $T \nleq t$

While $slk(A, b) \neq 0$

Find r s.t. $b_r - (A \otimes (A^* \otimes b))_r = slk(A, b)$.

Find s s.t.

$$slk(A, b(r, s)) = \min_{j=1,\dots,m} slk(A, b(r, j)).$$

If

$random(0, 1) < \exp \frac{slk(A,b)-slk(A,b(r,s))}{T}$

Set $\pi_{z+1} := \pi_z(r, s)$, $z := z + 1$ and $T := rT$.

else

$slk(A, b)$ cannot be improved anymore. Therefore

$$\bar{\pi} := \min_z(\pi_z)$$

and $slk(A, b(\bar{\pi}))$ is the resulting slack. Stop.

$slk(A, b(\pi_z)) = 0$ implies that $b(\pi_z)$ is an image of A and therefore $\bar{\pi} := \pi_z$ and $slk(A, b(\bar{\pi_z})) = 0$.

Termination criteria is satisfied therefore

$$\bar{\pi} := \min_z(\pi_z)$$

and $slk(A, b(\bar{\pi}))$ is the resulting slack.

8.6 Test Results for Simulated Annealing

In this section we will run tests on the two algorithms we obtained by considering simulated annealing. We will use the same 20 sets of problems as before and we will also use the seven permutations we used before as our starting permutations. The results are as follows: The following table represents results obtained by using Simulated Annealing Full Local Search Method:

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
1	10	2	2131	Identity	2392	45	$4s$
				Atimes0	2232	42	$4s$
				Random 1	2392	45	$4s$
				Random 2	2392	45	$4s$
				Random 3	2392	45	$4s$
				Random 4	2392	45	$4s$
				Random 5	2392	45	$4s$
2	50	2	2343	Identity	4679	45	$7m38s$
				Atimes0	9695	45	$7m35s$
				Random 1	8499	45	$7m34s$
				Random 2	8704	45	$7m42s$
				Random 3	7100	45	$7m36s$
				Random 4	7688	42	$6m57s$
				Random 5	8841	45	$7m37s$
3	100	2	1503	Identity	9932	45	$1h12m13s$
				Atimes0	10898	45	$1h13m29s$
				Random 1	10486	45	$1h12m26s$
				Random 2	10912	45	$1h12m12s$
				Random 3	11051	43	$1h9m39s$
				Random 4	10096	45	$1h12m57s$
				Random 5	10652	45	$1h12m48s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
4	200	2	892	Identity	N/A		
				Atimes0			
				Random 1			
				Random 2			
				Random 3			
				Random 4			
				Random 5			
5	10	10	0	Identity	305	45	$6s$
				Atimes0	54	45	$6s$
				Random 1	113	29	$3s$
				Random 2	1	19	$1s$
				Random 3	1	22	$1s$
				Random 4	82	35	$4s$
				Random 5	44	19	$1s$
6	50	10	0	Identity	760	45	$16m55s$
				Atimes0	1679	44	$16m52s$
				Random 1	1933	45	$16m37s$
				Random 2	1827	45	$16m27s$
				Random 3	1743	45	$16m28s$
				Random 4	1760	45	$16m33s$
				Random 5	1674	45	$16m30s$
7	100	10	0	Identity	1514	45	$3h25m18s$
				Atimes0	3729	45	$3h29m51s$
				Random 1	3527	45	$3h26m31s$
				Random 2	3398	44	$3h30m12s$
				Random 3	2809	43	$3h20m17s$
				Random 4	3270	43	$3h21m5s$
				Random 5	3393	45	$3h20m44s$
8	200	10	0	Identity	N/A		
				Atimes0			
				Random 1			
				Random 2			
				Random 3			
				Random 4			
				Random 5			
9	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
10	50	50	0	Identity	93	45	$1h3m6s$
				Atimes0	256	42	$55m4s$
				Random 1	294	43	$1h2m34s$
				Random 2	280	45	$1h3m41s$
				Random 3	274	43	$1h14s$
				Random 4	288	45	$1h3m21s$
				Random 5	283	44	$1h37s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
11	100	50	0	Identity	N/A		
				Atimes0			
				Random 1			
				Random 2			
				Random 3			
				Random 4			
				Random 5			
12	200	50	0	Identity	N/A		
				Atimes0			
				Random 1			
				Random 2			
				Random 3			
				Random 4			
				Random 5			
13	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
14	50	100	0	Identity	0	22	$26m41s$
				Atimes0	42	43	$1h51m53s$
				Random 1	64	44	$1h54m54s$
				Random 2	90	38	$1h24m12s$
				Random 3	68	42	$1h45m14s$
				Random 4	85	40	$1h35m33s$
				Random 5	87	43	$1h49m44s$
15	100	100	0	Identity	N/A		
				Atimes0			
				Random 1			
				Random 2			
				Random 3			
				Random 4			
				Random 5			
16	200	100	0	Identity	N/A		
				Atimes0			
				Random 1			
				Random 2			
				Random 3			
				Random 4			
				Random 5			
17	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
18	50	200	0	Identity	0	2	$14s$
				Atimes0	0	6	$3m29s$
				Random 1	0	8	$6m30s$
				Random 2	0	4	$1m24s$
				Random 3	0	4	$1m24s$
				Random 4	0	2	$14s$
				Random 5	0	2	$14s$
19	100	200	0	Identity	N/A		
				Atimes0			
				Random 1			
				Random 2			
				Random 3			
				Random 4			
				Random 5			
20	200	200	0	Identity	N/A		
				Atimes0			
				Random 1			
				Random 2			
				Random 3			
				Random 4			
				Random 5			

Table 8.13: Results obtained using Simulated Annealing Full Local Search Method for 20 matrices with different dimensions.

The following table represents results obtained by using Simulated Annealing Semi-full Local Search Method:

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
1	10	2	2131	Identity	3124	7	$< 1s$
				Atimes0	8018	2	$< 1s$
				Random 1	7498	3	$< 1s$
				Random 2	7019	3	$< 1s$
				Random 3	10122	2	$< 1s$
				Random 4	8317	3	$< 1s$
				Random 5	8134	2	$< 1s$
2	50	2	2343	Identity	13526	45	$18s$
				Atimes0	14410	2	$< 1s$
				Random 1	16160	1	$< 1s$
				Random 2	14720	2	$< 1s$
				Random 3	14015	1	$< 1s$
				Random 4	12648	3	$< 1s$
				Random 5	14141	2	$< 1s$
3	100	2	1503	Identity	12726	45	$1m31s$
				Atimes0	14798	2	$< 1s$
				Random 1	14387	3	$< 1s$
				Random 2	15053	3	$< 1s$
				Random 3	15622	1	$< 1s$
				Random 4	15935	3	$< 1s$
				Random 5	14579	1	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
4	200	2	892	Identity	16401	45	$7m57s$
				Atimes0	16973	1	$< 1s$
				Random 1	15280	2	$1s$
				Random 2	16005	3	$3s$
				Random 3	15935	2	$1s$
				Random 4	14856	1	$< 1s$
				Random 5	16369	3	$3s$
5	10	10	0	Identity	868	45	$1s$
				Atimes0	1012	2	$< 1s$
				Random 1	708	3	$< 1s$
				Random 2	1055	2	$< 1s$
				Random 3	1846	1	$< 1s$
				Random 4	1190	1	$< 1s$
				Random 5	587	3	$< 1s$
6	50	10	0	Identity	2262	45	$39s$
				Atimes0	2768	3	$< 1s$
				Random 1	2827	4	$< 1s$
				Random 2	2936	3	$< 1s$
				Random 3	2976	1	$< 1s$
				Random 4	2692	2	$< 1s$
				Random 5	2809	2	$< 1s$
7	100	10	0	Identity	3857	45	$4m4s$
				Atimes0	4598	4	$2s$
				Random 1	4590	3	$2s$
				Random 2	4749	2	$< 1s$
				Random 3	4706	2	$< 1s$
				Random 4	4487	2	$< 1s$
				Random 5	4724	2	$< 1s$
8	200	10	0	Identity	5707	45	$28m13s$
				Atimes0	5050	2	$5s$
				Random 1	4965	2	$5s$
				Random 2	5185	1	$2s$
				Random 3	4786	2	$5s$
				Random 4	5121	3	$10s$
				Random 5	4888	4	$17s$
9	10	50	0	Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
				Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
10	50	50	0	Identity	500	45	$2m24s$
				Atimes0	533	5	$2s$
				Random 1	522	3	$< 1s$
				Random 2	463	2	$< 1s$
				Random 3	603	5	$2s$
				Random 4	675	1	$< 1s$
				Random 5	526	1	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
				Identity	859	45	$19m1s$
				Atimes0	895	4	$12s$
				Random 1	984	3	$7s$
11	100	50	0	Random 2	967	4	$12s$
				Random 3	1049	3	$7s$
				Random 4	1002	2	$4s$
				Random 5	1219	1	$1s$
				Identity	1085	45	$2h9m56s$
				Atimes0	1155	4	$1m38s$
				Random 1	1096	3	$47s$
12	200	50	0	Random 2	1275	2	$24s$
				Random 3	1162	6	$2m47s$
				Random 4	1117	5	$1m59s$
				Random 5	1206	2	$24s$
				Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
13	10	50	0	Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$
				Identity	214	45	$4m49s$
				Atimes0	203	4	$3s$
				Random 1	153	6	$6s$
14	50	100	0	Random 2	383	1	$< 1s$
				Random 3	217	2	$< 1s$
				Random 4	237	3	$2s$
				Random 5	124	2	$< 1s$
				Identity	519	45	$32m59s$
				Atimes0	419	2	$6s$
				Random 1	408	3	$12s$
15	100	100	0	Random 2	639	2	$6s$
				Random 3	412	8	$1m10s$
				Random 4	578	2	$6s$
				Random 5	470	3	$12s$
				Identity	679	45	$4h6m11s$
				Atimes0	777	6	$5m13s$
				Random 1	704	3	$1m30s$
16	200	100	0	Random 2	735	1	$15s$
				Random 3	679	4	$2m29s$
				Random 4	809	4	$2m29s$
				Random 5	716	3	$1m39s$
				Identity	0	0	$< 1s$
				Atimes0	0	0	$< 1s$
				Random 1	0	0	$< 1s$
17	10	50	0	Random 2	0	0	$< 1s$
				Random 3	0	0	$< 1s$
				Random 4	0	0	$< 1s$
				Random 5	0	0	$< 1s$

Test matrix	m	n	Optimal	Starting permutation	Result	Step	Time
18	50	200	0	Identity	12	45	$8m55s$
				Atimes0	56	1	$< 1s$
				Random 1	55	3	$3s$
				Random 2	0	4	$4s$
				Random 3	30	3	$3s$
				Random 4	0	2	$< 1s$
				Random 5	0	2	$< 1s$
19	100	200	0	Identity	286	45	$1h4m31s$
				Atimes0	212	2	$12s$
				Random 1	320	3	$23s$
				Random 2	292	3	$23s$
				Random 3	251	2	$12s$
				Random 4	255	6	$1m12s$
				Random 5	274	2	$12s$
20	200	200	0	Identity	408	45	$8h21m56s$
				Atimes0	351	10	$28m24s$
				Random 1	369	7	$14m18s$
				Random 2	353	2	$1m32s$
				Random 3	409	2	$1m32s$
				Random 4	400	6	$10m35s$
				Random 5	388	4	$5m6s$

Table 8.14: Results obtained using Simulated Annealing Semi-full Local Search Method for 20 matrices with different dimensions.

The following table compares the results obtained from the two simulated annealing methods:

Test matrix	m	n	Optimal	Starting permutation	SA-FLS	SA-SFLS
1	10	2	2131	Identity	2392	3124
				Atimes0	2232	8018
				Random 1	2392	7498
				Random 2	2392	7019
				Random 3	2392	10122
				Random 4	2392	8317
				Random 5	2392	8134
2	50	2	2343	Identity	4679	13526
				Atimes0	9695	14410
				Random 1	8499	16160
				Random 2	8704	14720
				Random 3	7100	14015
				Random 4	7688	12648
				Random 5	8841	14141
3	100	2	1503	Identity	9932	12726
				Atimes0	10898	14798
				Random 1	10486	14387
				Random 2	10912	15053
				Random 3	11051	15622
				Random 4	10096	15935
				Random 5	10652	14579

Test matrix	m	n	Optimal	Starting permutation	SA-FLS	SA-SFLS
4	200	2	892	Identity	N/A	16401
				Atimes0		16973
				Random 1		15280
				Random 2		16005
				Random 3		15932
				Random 4		14856
				Random 5		16369
5	10	10	0	Identity	305	868
				Atimes0	54	1012
				Random 1	113	708
				Random 2	1	1055
				Random 3	1	1846
				Random 4	82	1190
				Random 5	44	587
6	50	10	0	Identity	760	2262
				Atimes0	1679	2768
				Random 1	1933	2827
				Random 2	1827	2936
				Random 3	1743	2976
				Random 4	1760	2692
				Random 5	1674	2809
7	100	10	0	Identity	1514	3857
				Atimes0	3729	4598
				Random 1	3527	4590
				Random 2	3398	4749
				Random 3	2809	4706
				Random 4	3270	4487
				Random 5	3393	4724
8	200	10	0	Identity	N/A	5707
				Atimes0		5050
				Random 1		4965
				Random 2		5185
				Random 3		4786
				Random 4		5121
				Random 5		4888
9	10	50	0	Identity	0	0
				Atimes0	0	0
				Random 1	0	0
				Random 2	0	0
				Random 3	0	0
				Random 4	0	0
				Random 5	0	0
10	50	50	0	Identity	93	500
				Atimes0	256	533
				Random 1	294	522
				Random 2	280	463
				Random 3	274	603
				Random 4	288	675
				Random 5	283	526

Test matrix	m	n	Optimal	Starting permutation	SA-FLS	SA-SFLS
11	100	50	0	Identity	N/A	859
				Atimes0		895
				Random 1		984
				Random 2		967
				Random 3		1049
				Random 4		1002
				Random 5		1219
12	200	50	0	Identity	N/A	1085
				Atimes0		1155
				Random 1		1096
				Random 2		1275
				Random 3		1162
				Random 4		1117
				Random 5		1206
13	10	50	0	Identity	0	0
				Atimes0	0	0
				Random 1	0	0
				Random 2	0	0
				Random 3	0	0
				Random 4	0	0
				Random 5	0	0
14	50	100	0	Identity	0	214
				Atimes0	42	203
				Random 1	64	153
				Random 2	90	383
				Random 3	68	217
				Random 4	85	237
				Random 5	87	124
15	100	100	0	Identity	N/A	519
				Atimes0		419
				Random 1		408
				Random 2		639
				Random 3		412
				Random 4		578
				Random 5		470
16	200	100	0	Identity	N/A	679
				Atimes0		777
				Random 1		704
				Random 2		735
				Random 3		679
				Random 4		809
				Random 5		716
17	10	50	0	Identity	0	0
				Atimes0	0	0
				Random 1	0	0
				Random 2	0	0
				Random 3	0	0
				Random 4	0	0
				Random 5	0	0

Test matrix	m	n	Optimal	Starting permutation	SA-FLS	SA-SFLS
18	50	200	0	Identity	0	12
				Atimes0	0	56
				Random 1	0	55
				Random 2	0	0
				Random 3	0	30
				Random 4	0	0
				Random 5	0	0
19	100	200	0	Identity	N/A	296
				Atimes0		212
				Random 1		320
				Random 2		292
				Random 3		251
				Random 4		255
				Random 5		274
20	200	200	0	Identity	N/A	408
				Atimes0		351
				Random 1		369
				Random 2		353
				Random 3		409
				Random 4		400
				Random 5		388

Table 8.15: Comparison of the results obtained from the two simulated annealing methods.

Unfortunately due to the limitations of the computers available we were not able to obtain all the results. In these experiments we have set the range of the matrices and vectors to be ten thousands. We have set $temperature = 100$, $probability = 0.5$ and the $decreasing\ ratio = 0.9$ for the results obtained. When either the dimension of m or n reaches one hundred, the computational time for the SA-FLS method started to increase dramatically. We were not able to obtain the results within 10 hours by using a 2.8GHz processor. These entries are labelled N/A in the tables. But by using results from SA-FLS, we can give a good estimate on the results that are not available.

From Table 8.13 and table14 we can see that both algorithms use considerable more time than FLS and S-FLS. But again SA-SFLS is much faster than SA-FLS.

If we compare the results obtained from the two algorithms by looking at Table 8.15, we can see that SA-FLS perform much better than SA-SFLS. Also we can see that when m is not significantly larger than n we are getting a good approximate to the optimal solution. But when m is significantly larger than n we are still unable to get a acceptable approximate to the optimal solution. We may need to develop some other forms of specified heuristic for the case when $m >> n$ to obtain a good approximate to the optimal solution.

8.7 Summary

In this chapter, we have discussed heuristics methods on solving the permuted linear system problem. We have developed three heuristic methods to obtain a approximation of the solution to the permuted linear systems problem. They are the full local search method, semi-full local search method and column maxima method.

Unfortunately we were not able to obtain a reasonable approximation from these three methods, therefore we have developed another two methods by considering simulated annealing with the local search methods we developed earlier. Henceforth we obtained simulated annealing full local search method and simulated annealing semi-full local search method.

From the tests we made, we found that the new methods give a better approximation than the three results we developed originally. But when the number of rows of the matrix is significantly larger than the number of columns, we were still unable to obtain a good heuristic solution with the new method.

Chapter 9

Conclusion and Future Research

9.1 Summary

In this thesis we have studied the concept of max-algebra and we have presented solution methods on max-algebraic linear system. We have considered the theory of max-algebraic eigenproblem and we have found that this have great interaction with synchronization of discrete event systems, i.e. machine scheduling and railway timetabling.

In the first chapter, we have presented the historical background and some of the major works done in the field of max-algebra. In Chapter 2 and 3, we have summarized some well known results in max-algebra. The results presented in Chapter 4, 5, 6, 7 and 8 are all original.

In Chapter 2 we have provided the terminology, notations and basic definitions of max-algebra. We have also presented some of the theories on linear system and we have shown that the linear systems can be formulated as a set covering problem. Using the definitions of linear system, we have defined the notions of image set and simple set which played an important part in this thesis.

In Chapter 3 we have presented definitions and results on max-algebraic eigenvalue-eigenvector problem. At first we have discussed the concept of steady state and how it is related to the max-algebraic eigenproblem. Then we have presented definitions and results on graph theory and we have shown that the max-algebraic eigenproblem is very much related to graph theory. Using this relation, we have presented a solution method on finding all eigenvalues and eigenvectors for any square matrices.

We have shown that the maximum cycle mean of a matrix plays an important role on solving the eigenproblem and we have shown that it always is an eigenvalue. It is called the principal eigenvalue. We have also seen that square matrix can have at most n distinct eigenvalues where n is the order of the matrix. We have presented results on finding all eigenvalues and the set of eigenvectors for each eigenvalue. Using this, we can generate at most n rectangular matrices; one for each eigenvalue, such that the set of eigenvectors can be obtained by considering the image set of these matrices. Henceforth the problem of optimizing eigenvectors has been transformed into optimizing the image set of a matrix.

In Chapter 4 we have considered the problem of minimizing and maximizing the range norm of an image of a matrix. In the case of minimization, we have shown that when the image is finite, the solution can be found by considering $A \otimes \bar{x}$ where $\bar{x}$ is the principal solution of the linear system $A \otimes x = 0$. We have generalized the result to the case when the image may not be finite and developed an algorithm to solve the problem of minimizing and maximizing the range norm for this case. For maximization, the solution is obtained by considering the range norm for each column of the matrix. We have shown that the solution is bounded only if the matrix is finite.

In Chapter 5, we have investigated the case of minimizing and maximizing the range norm of an image when some of the components of the vector are given and fixed. We have shown that both the minimization and the maximization problem are very similar to the counterpart in Chapter 4 for the case when only one component is prescribed. We have developed a method on solving the case when all but one component were prescribed and using this result, we were able to develop an algorithm to solve the general case. Furthermore we were also able to use the algorithm we developed for the minimization problem and modified it to solve the maximization case.

In Chapter 6, we have investigated the integer linear system problem. It turns out that the problem can be easily solved when the matrix only consists of one or two columns, i.e. $n = 1$ and $n = 2$. We were able to obtain a necessary and sufficient condition such that the matrix has an integer image for both of these cases.

We have also shown that for every square matrix, we can transform it into a strongly definite matrix without affecting its image set (or its simple image set if it is strongly regular). We have seen that for a strongly regular and strongly definite matrix, we are able to obtain the integer simple set of this matrix efficiently. We have seen that the integer simple image set is exactly the set of integer eigenvectors of another matrix which is obtained from A. Using this, we have found some sufficient conditions for an integer image to exist for any strongly definite matrix.

We have also investigated the general case. We have obtained a upper and lower bound between any two components for any integer image of a finite matrix. Using this result, we have developed an algorithm to generate all possible candidates for an integer image which provides us a benchmark on solving integer linear systems for any matrices in general.

In Chapter 7, we have investigated the permuted linear system problem which is NP-complete. We have developed algorithms to decide if a permuted vector is in the image of A for the case of $n = 2$, $n = 3$ and $n > 3$. We have shown that the problem is trivial when the matrix only contains one column. For the case when $n = 2$, we have developed an algorithm to solve it efficiently. For the case $n = 1$ and $n = 2$ we have also developed algorithms to find out if we can find a permuted vector such that the distance from this vector to the image set is minimized. We measured this distance by using the Chebyshev norm.

Finally in Chapter 8 we have developed three heuristic methods to obtain a approximation of the solution to the permuted linear systems problem. Unfortunately we were not able to obtain a reasonable approximation from this method therefore we have developed another two methods by using simulated annealing. Although we got a better approximation than before, the results still were not a good enough approximation when $m >> n$.

9.2 Possible Future Research

When we are investigating the range norm of the image set, we have only considered the case of one sided linear system, i.e. $A \otimes x = b$. It may be interesting to consider the case when we have a two-sided linear system, i.e, $A \otimes x = B \otimes y$ or $A \otimes x = B \otimes x$ and investigate the case of minimizing or maximizing the range norm of solutions to such systems. It may also be interesting to consider the case when some components are prescribed as well.

While investigating the integer linear system problem, we were not able to find an efficient algorithm for the general case. One of the interesting research directions for this problem is to either prove that it is unlikely that an efficient algorithm exists, i.e. NP-hard or NP-complete, or we find an efficient algorithm to solve this problem.

Another possible area for future research will be to combine the range norm problem with the integer linear systems problem. It is likely that in real-life situations, a manufacturer would prefer the range norm of the starting time vector or completion time vector to be minimized/maximized and at the same time consists of only integer values.

For the case of permuted linear systems problem, we were not able to obtain a good heuristic method for approximating an acceptable result for this problem. It may be interesting to investigate the structure of the matrix A and obtain some sufficient conditions for b to be a permuted image of A. One such condition would be if there exists a permutation such that when b is permuted, then it is equal to one of the column of A or a multiple of one of the column of A. If this condition is satisfied, then we can immediately say that b is an image of

A.

Appendix A

On some properties of the image set of a max-linear mapping

In this appendix a paper co-written by myself and Dr P. Butkovič entitled *On some properties of the image set of a max-linear mapping* [23] is presented. This paper is published in Contemporary Mathematics Series, AMS Providence, 495 (2009) 115-126.

List of References

[1] M. Akian, R. Bapat, S. Gaubert, Max-plus algebra, in: L. Hogben(Ed.), Handbook of Linear algebra: Discrete Mathematics and its Application, Chapman & Hall/CRC, (2007).

[2] M. Akian, S. Gaubert, V.N. Kolokoltsov, Set coverings and invertibility of functional Galois connections, Idemoptent mathematics and mathematical physics, 19-51, Contemp. Math., 377, Amer. Math. Soc., Providence, RI, (2005).

[3] M. Akian, S. Gaubert, C. Walsh, Discrete max-plus spectral theory, in Idempotent Mathematics and Mathematical Physics, Contemp. Math. 377, G. L. Litvinov and V. P. Maslov, eds., AMS, Providence, RI, 2005, pp. 5377.

[4] M. Akian, J.P. Quadrat, M. Viot, Duality between probability and optimization in: Gunawardena(Ed.), Idempotency, Cambridge, (1988) 331-353.

[5] N. Bacaër, Modèles mathématiques pour Poptimisation des rotations, Comptes Rendus de Académie d'Agriculture de France, 89(3):52 (2003).

[6] F.L. Bacelli, G. Cohen, G.J. Olsder, J.P. Quadrat, Synchronization and Linearity, An Algebra for Discrete Event Systems, Wiley, Chichester, (1992).

[7] R.B. Bapat, D. Stanford, P. van den Driessche, Pattern properties and spectral inequalities in max-algebra, Journal of Matrix Analysis and Applications 16(3) (1995), 964-976.

[8] R.B. Bapat, D. Stanford, P. van den Driessche, The eigenproblem in max algebra, DMS-631-IR, University of Victoria, British Columbia, 1993.

[9] A.Berman and R.J.Plemmons, Nonnegative matrices in the mathematical sciences, Academic Press, 1979

[10] P.A. Binding, H. Volkmer, A generalized eigenvalue problem in the max algebra, Linear Algebra & Appl. 422 (2007) 360-371.

[11] T. S. Blyth, M. F. Janowitz, Residuation Theory, Pergamon press (1972).

[12] R. Bronson, G. Naadimuthu, Schaums' Outline Theory and Problems of Operations Research, McGraw-Hill, (1997).

[13] R.E.Burkard and E. ela, Linear assignment problems and extensions. Handbook of combinatorial optimization, Supplement Vol. A, 75?149, Kluwer Acad. Publ., Dordrecht, 1999.

[14] P. Butkovič, Max-algebra:the linear algebra of combinatorics?, Linear Algebra & Appl. 367 (2003) 313-335.

[15] P. Butkovič, Strong regularity of matrices- a survey of results, Discrete Applied Mathematics, North-Holland, 48 (1994), 45-68.

[16] P. Butkovič, Permuted max-algebraic eigenvector problem is NP-complete, Linear Algebra & Appl. 428 (2008), 1874-1882.

[17] P. Butkovič, R. A. Cuninghame-Green An $O(n^2)$ algorithm for the maximum cycle mean of an $n \times n$ bivalent matrix, Discrete Appl. Math., 35 (1992) 157-162.

[18] P. Butkovič, R.A. Cuninghame-Green, Bases in max-algebra, Linear Algebra & Appl. 389 (2004), 107-120.

[19] R.A. Cuninghame-Green, P. Butkovič, Generalised eigenproblem, The University of Birmingham, preprint 2008/12.

[20] P. Butkovič, R.A. Cuninghame-Green, On matrix powers in max-algebra, Linear Algebra & Appl. 421 (2007), 370–381.

[21] P. Butkovič, R.A. Cuninghame-Green, S. Gaubert, Reducible spectral theory with applications to the robustness of matrices in max-algebra, SIAM Journal on Matrix Analysis and Applications 31(3) (2009) 1412-1431.

[22] P. Butkovič, G. Hegedüs, An elimination method for finding all solutions of the system of linear equations over an extremal algebra, Ekonom. mat. Obzor 20 (1984) 2003-215.

[23] P. Butkovič, K.P.Tam, On some properties of the image set of a max-linear mapping, Contemporary Mathematics Series, AMS Providence, 495 (2009) 115-126.

[24] Z.Q. Cao, K.H. Kim, F.W. Roush, Incline algebra and applications, Ellis Horwood (1984).

[25] J.P. Comet, Application of Max-plus algebra to biological sequencing comparisons, Theoret. Comput. Sci., 293 (2003) 189-217.

[26] A. Dasdan, R.K. Gupta, Faster maximum and minimum mean cycle algorithms for system-performance analysis, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 17(10):889–899, October (1998).

[27] R.A.Cuninghame-Green, Describing industrial processes with interference and approximating their steady-state behaviour, Operations Research Quarterly 13(1962) 95-100

[28] R.A. Cuninghame-Green, Projections in minimax algebra. Math. Programming 10 (1976), no. 1, 111-123.

[29] R.A. Cuninghame-Green, Minimax Algebra, Lecture Notes in Economics and Mathematical Systems, vol.166, Springer, Berlin, (1979).

[30] R.A. Cuninghame-Green, Minimax Algebra and applications, in: Advances in Imaging and Electron Physics, vol. 90, Academic Press, New York, (1995), 1-121.

[31] R.A. Cuninghame-Green, Process synchronisation in a steelworks - a problem of feasibility. In Proc. 2nd Int. Conf. on Operational Research (ed. by Banburry and Maitland), English University Press, (1960) pp. 323-328.

[32] R.A. Cuninghame-Green, P. Butkovič, The equation $A \otimes x = B \otimes y$ over $(max, +)$, Theoret. Comput. Sci. 293 (1991) 3-12.

[33] R.A. Cuninghame-Green, K. Zimmermann, Equation with residual functions, Comment. Math. Uni. Carolina. 42(2001) 729-740.

[34] L. Elsner, P. van den Driessche, On the power method in max algebra, Linear Algebra & Appl. 302-303 (1999) 17-32.

[35] L. Elsner, P. van den Driessche, Modifying the power method in max algebra, Linear Algebra & Appl. 332-334 (2001) 3-13.

[36] M.R. Garey, D.S. Johnson, Computers and intractability, A guide to the theory of NP-completeness, Bell Labs, (1979).

[37] S. Gaubert, Methods and Application of (max,+) Linear Algebra, INRIA, (1997).

[38] S. Gaubert, Théorie des systèmes linéaires dans les dioïdes, Theèse, Ecole des Mines de Paris, 1992.

[39] M. Gondran, M. Minoux, Graphes et algorithmes, Eyrolles, Paris, (1979). Engl. transl. Graphs and Algorithms, Wiley (1984).

[40] M. Gondran, M. Minoux, Linear algebra in dioids: a survey of recent results, Annals of Discrete Mathematics,19 (1984) 147-164.

[41] M. Gondran, M. Minoux, Valeurs propres et vecteur propres dans les dioïdes et leur inteprétation en théorie des graphes, Bulletin de la direction des etudes et recherches, Serie C, Mathematiques et Informatiques, No 2, 1977 (25-41).

[42] K. Hashiguchi, Improved limitedness theorems on finite automata with distance functions, Theoret. Comput. Sci., 72 (1990) 27-38.

[43] B. Heidergott, G. J. Olsder, J. van der Woude, Max-plus at work, Princeton University Press, New Jersey, (2006).

[44] R.M. Karp. A characterization of the minimum cycle mean in a digraph, Discrete Mathematics 23, 309-311, (1978).

[45] V.N. Kolokoltsov, Idempotent analysis and its applications, Kluwer Academic Publishers Groups, Dordrecht, (1997).

[46] D. Krob, A. Bonnier-Rigny, A complete system of identities for one letter rational expressions with multiplicities in the tropical semiring, J. Pure Appl. Algebra, 134 (1994) 27-50.

[47] H. Leung Limitedness theorem on finite automata with distance function: an algebraic proof, Thoret. Comput. Sci., 81 (1991) 137-145.

[48] S. Lipschutz, M. Lipson, Schaums' Outline Theory and Problems of Linear Algebra, McGraw-Hill, (2001).

[49] V.P Maslov, Méthodes Operatorielles, Mir, Moscou, (1973).

[50] Z. Michalewicz, D. B. Fogel, How to Solve It: Modern Heuristics Second Edition Springer, (2004).

[51] P.D. Moral, Maslov optimization theory: topological aspects in: Gunawardena(Ed.), Idempotency, Cambridge, (1988) 354-382.

[52] P.D. Moral, G. Salut, Random particle methods in (max,+) optimization problems in: Gunawardena(Ed.), Idempotency, Cambridge, (1988) 383-391.

[53] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization-Algorithms and Complexity, Dover, (1998).

[54] A.P.Punnen and K.P.K.Nair, Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem, Discrete Appl. Math. 55(1) (1994), 91-93.

[55] J.E. Pin, Tropical semirings, in: Gunawardena(Ed.), Idempotency, Cambridge, (1988) 50-69.

[56] I.V. Romanovski, Optimization and stationary control of discrete deterministic process in dynamic programming. Kibernetika, 2:66-78, 1967. Engl. transl. in Cybernetics 3 (1967).

[57] K.H.Rosen et al, Handbook of discrete and combinatorial mathematics, CRC Press 2000.

[58] S. Sergeev, Alternating method for homogeneous systems of equations over max algebra, University of Birmingham-Preprint 2008/18, (2008).

[59] I. Simon, Limited subsets of free monoid, 19th Annual Symposium on Foundations of Computer Science, IEEE, (1978) 143-150.

[60] I. Simon, On semigroups of matrices over the tropical semiring, Theoret. Infor. and Appl, 28(3-4) (1994) 277-294.

[61] H.Schneider, The Influence of the Marked Reduced Graph of a Nonnegative Matrix on the Jordan Form and on Related Properties: A Survey, Linear Algebra and Its Applications 84:161-189 (1988).

[62] K.P. Tam, Optimising and approximating eigenvectors in max-algebra, MPhil(Qual) Thesis, The University of Birmingham, (2008).

[63] N.N. Vorobjov, Extremal algebra of positive matrices, Elektronische Datenverarbeitung und Kybernetik 3 (1967) 39-71. (in Russian)

[64] E.A. Walkup, G. Boriello, A general linear max-plus solution technique, in: Gunawardena(Ed.), Idempotency, Cambridge, (1988) 406-415.

[65] L.A. Wolsey, Integer programming, Chichester : Wiley, (1998).

[66] K. Zimmermann, Extremální algebra, Výzkumná publikace Ekonomicko-matematickélaboratoře při Ekonomickém ústave ČSAV, 46, Praha, (1976) (in Czech).

[67] U. Zimmermann, Linear and combinatorial optimization in ordered algebraic structures, Ann. Discrete Math. 10 (1981).