UNIFORM EMBEDDING OF ROBINSON SIMILARITY MATRICES

by

Zhiyuan Zhang

Submitted in partial fulfillment of the requirements
for the degree of Master of Science

at

Dalhousie University
Halifax, Nova Scotia
April 2021

Under the impact of COVID-19
2019–present

Table of Contents

List of Figures

Abstract

A Robinson similarity matrix is a symmetric matrix where the entry values on all rows and columns increase toward the diagonal. Decompose the Robinson matrix into the sum of k $\{0, 1\}$-matrices, then these k $\{0, 1\}$-matrices are the adjacency matrices of a set of nested unit interval graphs. Previous studies show that unit interval graphs coincide with indifference graphs. An indifference graph has an embedding that maps each vertex to a real number, where two vertices are adjacent if their embedding is within a fixed threshold distance. In this thesis, we consider k different threshold distances and study the problem of finding an embedding that, *simultaneously* and with respect to each threshold distance, embeds the k indifference graphs corresponding to the k adjacency matrices. This is called a *uniform embedding* of a Robinson matrix with respect to the k threshold distances. We give a sufficient and necessary condition on Robinson matrices that have a uniform embedding, which is derived from paths in an associated graph. We also give an efficient combinatorial algorithm to find a uniform embedding or give proof that it does not exist, for the case where $k = 2$.

List of Abbreviations and Symbols Used

G	graph
$V = V(G)$	the vertex set of graph G
$E = E(G)$	the edge set of graph G
$n = \|V(G)\|$	the order of graph G and the dimension of square matrix
$[n]$	the set of natural number from 1 to n
k	dimension of threshold vectors
$A = (a_{i,j})_{i,j\in[n]}$	$n \times n$ matrix with real entry $a_{i,j}$
$\mathcal{S}^n[k]$	the set of $n \times n$ Robinson matrices
$\boldsymbol{d} = (d_i)_{i\in[k]}$	k-dimensional real column vector with entry d_i
$\mathbb{D}_k$	set of all monotone decreasing vector of dimension k
$\boldsymbol{\chi}_t$	one-hot k-dimensional vector with 1 on index t
$\mathbf{0}$	k-dimensional zero vector
$\mathbf{1}$	k-dimensional all-one vector
$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{x}, \boldsymbol{y}, \ldots$	vectors
u, v, u_-, v_-	vertices

Acknowledgements

First, I would like to thank Dr. Jeannette Janssen for being my supervisor and coming up with this interesting project. Her encouragement and kindness are great supports; I could not build up this thesis without her guidance. I would also thank the readers of this thesis, Dr. Jason Brown and Dr. Nauzer Kalyaniwalla, for reading this thesis and providing suggestions. I had opportunities to take their courses in different stages over my study in math. It is an honour to be examined by the professors who opened up my horizon in mathematics.

It is also worth mentioning that this thesis is done during the COVID-19 pandemic spreads around the globe. I want to thank everyone who coordinates things over this particular period and keep things running so that we, students, could finish our degree without serious impact. I want to thank my friends: it is always nice to talk to someone, online though, when the city has a lockdown. Finally, I would like to thank my family for understanding and supporting me in pursuing my graduate studies.

Halifax, 2021

Chapter 1

Introduction

1.1 Definitions and Notations

We lay down definitions and notations to help the narration. Let $[n] = \{1, 2, \ldots, n\}$ denote the set of natural numbers from 1 to n. Extend this notation on 0 and define $[0] = \{\}$. A permutation τ is a bijection on $[n]$. Denote the set of all permutations on $[n]$ as $\mathcal{P}_n$. We will write the permutation function in the form of a sequence, $\tau = (\tau_1, \ldots, \tau_n)$, for $\tau_i \in [n]$ where $\tau_i = \tau(i)$. The *reversal* of a permutation τ is a permutation τ' such that $\tau'(i) = \tau(n - i)$. In other word, the reversal of permutation $\tau = (\tau_1, \ldots, \tau_n)$ is a permutation defined as $\tau' = (\tau_n, \ldots, \tau_1)$. We use $[a, b]$ and (a, b) to denote closed and open real intervals, respectively.

A function f defined on an ordered set S, for $i, j \in S$, f is *monotonically increasing* if $i \leqslant j \iff f(i) \leqslant f(j)$ and the set is *monotonically decreasing* if $i \leqslant j \iff f(i) \geqslant f(j)$. The function f is *monotone* if it is either monotonically increasing or monotonically decreasing. The function f is *strictly increasing* if $i < j \iff f(i) < f(j)$ and is *strictly decreasing* if $i < j \iff f(i) > f(j)$. We extend the increasing or decreasing property to any sequence, ordered set, and vectors. For example, vector $\boldsymbol{x} = (x_i)$ is strictly decreasing if $x_i > x_{i+1}$.

A *graph* $G = (V, E)$ consists of a vertex set and an edge set, where $V = \{v_i : i \in [n]\}$ and any edge $e \in E$ is of the form $e = \{v_i, v_j\}$.

Write an $m \times n$ real-valued matrix A in the form of $(a_{i,j})_{i\in[m],j\in[n]}$ where $a_{i,j} \in \mathbb{R}$. We will omit the subscript if the dimension is clear and write $A = (a_{i,j})$. Matrix A is *symmetric* if A is a square matrix, say $n \times n$, and $a_{i,j} = a_{j,i}$ for all $i, j \in [n]$. The *transpose* of an $n \times m$ matrix A is an $m \times n$ matrix, denoted as $A^\top = (a'_{j,i})_{j\in[m],i\in[n]}$, has $a_{i,j} = a'_{j,i}$ for $i \in [m], j \in [n]$. We denote I_n as the $n \times n$ identity matrix, and J_n as the $n \times n$ all-one matrix. A matrix is a *diagonal matrix* if the only non-zero entries are on the diagonal, *i.e.*, $a_{i,j} \neq 0$ implies $i = j$. Let $\tau \in \mathcal{P}_n$, the *permutation matrix of* τ, $T = (p_{i,j})_{i,j\in[n]}$, is a binary matrix with $p_{i,\tau(i)} = 1$ and 0 otherwise. Applying T and its transpose to both sides of a matrix, $TAT^\top = (a_{\tau(i),\tau(j)})$, we obtain a matrix reordered by τ, denoted as $A^\tau = TAT^\top$.

Matrices are binary if the entries are taken from either 0 or 1. Given graph $G = ([n], E)$, an $n \times n$ symmetric binary matrix A with $A = (a_{i,j})$ is the *adjacency matrix of* G if $a_{i,j} = 1 \iff \{i,j\} \in E$, otherwise $a_{i,j} = 0$. Conversely, let A be an $n \times n$ binary matrix, then G is the *graph of matrix* A if $G = ([n], E)$ with $a_{i,j} = 1 \iff \{i,j\} \in E$. More generally, we say matrix $A = (a_{i,j})$ is a *generalized adjacency matrix of graph* G if $a_{i,j} \neq 0 \iff \{i,j\} \in E$.

Vertex u is a *neighbour* of vertex v if $\{u,v\} \in E$. The *neighbour set of* v, $N(v)$, is the set of all neighbours of v. Define the *closed neighbour set of* v as $N[v] = \{v\} \cup N(v)$. A *clique* C is a subset of vertex V where all vertices in C are pair-wise adjacent to each other. A *maximal clique* is a clique which is not contained in another clique.

Two vertices are *undistinguishable* if they have the same closed neighbour set, *i.e.*, $N[u] = N[v]$ if u, v are distinguishable. In the sense of adjacency matrix or generalized adjacency matrix, $(a_{i,j})$, two rows u, v are *repeating rows* if $a_{u,j} = a_{v,j}$ for all $u \neq j \neq v$ and $a_{u,v} = a_{v,u} = a_{u,u} = a_{v,v}$.

A *subgraph of graph* G is a graph $G' = (V', E')$ where $V' \subseteq V$, and $E' \subset E$ for any $e = \{v_i, v_j\} \in E'$, we have that $v_i, v_j \in V'$. An *induced subgraph of graph* G, $G' = (V', E')$, is a subgraph of G and $e = \{v_i, v_j\} \in E' \iff e \in E$ and $v_i, v_j \in V'$.

A sequence of vertices in G, $W = \langle w_0, \ldots, w_p \rangle$, is a *walk* of length p if $\{w_{i-1}, w_i\} \in E$ for all $i \in [p]$: we also write it as an alternating sequence of vertices and edges, $W = \langle w_0, e_1, \ldots, w_p \rangle$, where $e_i = \{w_{i-1}, w_i\}$. A *cycle* of length c is a walk $W = \langle w_0, \ldots, w_c \rangle$ where $w_0 = w_c$ but there are no other repeated vertices in W. A *path* is a walk which contains no repeated vertices. An *intermediate vertices* of a path $W = \langle w_0, \ldots, w_p \rangle$ is any vertex w_i for $0 < i < p$. A graph is *connected* if for all pairs of vertices u and v, there exist a path contains both u, v. A *connected component* is a maximal connected subgraph, *i.e.*, no other subgraph properly contains it.

We use the definition in [1] to define the *irreducibility* of a matrix. A matrix A is *reducible* if there is permutation τ on $[n]$ so that

$$A^\tau = \left[\begin{array}{c|c} A_1 & 0 \\ \hline 0 & A_2 \end{array}\right]. \tag{1.1.1}$$

A matrix is irreducible if there is no such permutation exists.

An *interval graph* is a graph with a set of line segments in $\mathbb{R}$ (compact real intervals) as its vertex set, and there is an edge whenever two vertices intersect. An interval graph is *proper* if there is no vertex properly contains another vertex. A *unit interval graph* is an interval graph with all intervals having a unitary length.

An *indifference graph* $G = ([n], E)$ is a graph equipped with an embedding $\Pi : [n] \to \mathbb{R}$ and a *threshold distance* $d \in \mathbb{R}$, the vertices $u, v \in [n]$ are adjacent if and only if $|\Pi(v) - \Pi(u)| \leqslant d$.

A graph G is a *complete graph* with n vertices, denoted as K_n, if every pair of vertices are adjacent. A *complete bipartite graph* with $n + m$ vertices, denoted as $K_{m,n}$, is a graph with vertices partitioned into two sets V_1, V_2 with size m, n, where any pair of vertices from the same partition are not adjacent and any pair of vertices from different partitions are adjacent.

The *vertex-edge incidence matrix* of a graph G, $B = (b_{i,j})$, is a binary $m \times n$ matrix, with $n = |V|$ and $m = |E|$, where each column j corresponds to a vertex v_j and each row i corresponds to an edge e_i. The entry $b_{i,j} = 1$ if $e_i \in E$ and $v_j \in e$; $b_{i,j} = 0$ otherwise. The *clique-vertex incidence matrix* of a graph, $B = (b_{i,j})$, is a binary $c \times n$ matrix, with c is the number of different maximal cliques in the graph, where each row to a maximal clique C_i. Each entry $b_{i,j} = 1$ if $v_j \in C_i$; $b_{i,j} = 0$ otherwise.

A partial order is a binary relation, say $\preccurlyeq$, on a set S where

1. $\preccurlyeq$ is reflexive, *i.e.*, for all $a \in S$, $s \preccurlyeq s$;
2. $\preccurlyeq$ is antisymmetric, *i.e.*, for all $a, b \in S$, if $a \preccurlyeq b$ and $b \preccurlyeq a$, then $a = b$;
3. $\preccurlyeq$ is transitive, *i.e.*, for all $a, b, c \in S$, if $a \preccurlyeq b$ and $b \preccurlyeq c$, then $a \preccurlyeq c$.

1.2 Robinson matrix

Many research problems involve admitting a linear order on a set of pair-wise comparable items, together with a pair-wise similarity measure of these items, and the items are ordered closer if they are more similar than the farther pairs. There are several applications of this strategy in different fields, such as evolutionary biology, sociology, text mining, and visualization. Liiv ([16]) provided a thorough discussion on this topic. Among all applications, we consider archaeology as the classic example, where the items are the relics with undetermined manufacture date (year, era). We consider the relics to be ordered according to their ages. When the similarities between each pair are presented in the form of a matrix, this matrix will have the property that entries on each row and column increase toward the diagonal and decrease away from the diagonal. Such a matrix is called a *Robinson matrix*, or a Robinson *similarity* matrix.

Formally, write the matrix entry on the ith row and jth column as $(a_{i,j})$. A symmetric matrix is a Robinson matrix if

$$\begin{aligned} &\text{for } u < v < w \quad a_{u,v} \geqslant a_{u,w}, \\ &\text{for } v < w < u \quad a_{w,u} \geqslant a_{v,u}. \end{aligned} \tag{1.2.1}$$

1.3 The Problem

In his book [8], Doran described two events in archaeology that expose archaeologists to science and math. The first event is the development of the methods of *absolute dating*, such as radiocarbon dating of objects, which introduced science into archaeology. The second event is the invention of *sequence dating* by Flinders Petrie, which introduces mathematics to archaeology. Namely, absolute dating determines an "absolute" manufacture date of an object, with some scientific methods. In contrast, sequence dating only provides a linear order that reflects all the pairwise similarity measures to order the more similar pairs of objects closer to each other. The Robinson matrix is closely related to the relative dating problem. For instance, if we set the similarities as real numbers and the pairwise similarities to form a matrix, then the resulting matrix has the form of Robinson matrix (*i.e.*, property (1.2.1)) if the order reflects the actual chronology.

Example 1.1. Consider a set of objects $\{1, 2, 3, 4, 5\}$ and their manufacture dates, $\Pi(i)$, in term of years:

Object i	1	2	3	4	5
Manufacture date $\Pi(i)$	1900	1905	1906.5	1911.75	1912.75

We assume that the similarities $a_{i,j}$ are determined by the gap between the manufacture dates such that, for two threshold distances, 8 year and 6 years,

$$\begin{aligned} a_{u,v} = 2 &\iff 0 \leqslant \Pi(v) - \Pi(u) < 6, \\ a_{u,v} = 1 &\iff 6 < \Pi(v) - \Pi(u) < 8, \\ a_{u,v} = 0 &\iff 8 < \Pi(v) - \Pi(u). \end{aligned}$$

We obtain the similarity matrix $A = (a_{i,j})$ as

$$(a_{i,j}) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 2 & 2 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 1 & 2 & 2 \end{pmatrix} \end{array},$$

which is a Robinson matrix.

In Example 1.1, we compute a Robinson matrix from an absolute dating, and a solution to a sequence dating follows immediately. In this thesis, we ask the converse question, that is, **if we are given a solution of a sequence dating problem based on a set of similarities, can we find solutions to their absolute dating that satisfies all the similarities?** We call this absolute dating of the objects a *uniform embedding.*

Suppose we are given a set of objects with their similarities, and its relative dating problem is solved in the form of a Robinson matrix; for such a situation, we assume the similarities cannot be determined precisely, and give a "degree of t" of similarity between each pairs, where there are $k+1$ different degrees, $t = 0$ or $t \in [k]$, where k represents "very similar" and 0 represents "not similar at all". So each matrix is with integer entries $0 \leqslant a_{i,j} \leqslant k$. Let $\mathcal{S}_n[k]$ denote the set of all $n \times n$ Robinson matrices with entries that are taken from $[k]$ and the diagonal entries are all k and let $\mathbb{D}_k$ be the set of possible *threshold vectors*, $\mathbb{D}_k = \{\boldsymbol{d} \in \mathbb{R}^k : \boldsymbol{d} = (d_i)_{i\in[k]}, d_1 > \cdots > d_k > 0\}$. In this thesis, we define uniform embedding as the following.

Definition 1.2. Let $A = (a_{i,j})$ be a Robinson matrix in $\mathcal{S}^n[k]$. Given a threshold vector $\boldsymbol{d} \in \mathbb{D}_k$, then a map $\Pi : [n] \to \mathbb{R}$ is a *uniform embedding of A with respect to* $\boldsymbol{d}$ if, for each pair $u, v \in [n]$:

$$a_{u,v} = t \iff d_{t+1} < |\Pi(v) - \Pi(u)| \leqslant d_t \quad \text{for } t \in \{0, \ldots, k\}, \tag{1.3.1}$$

where we define $d_{k+1} = -\infty$ and $d_0 = \infty$, so that the lower bound for $a_{u,v} = k$ and the upper bound for $a_{u,v} = 0$ are trivially satisfied.

The recognition of Robinson matrices is extensively studied. Therefore, we assume the matrices in this thesis are in the form of Robinson matrices: we will discuss further in Section 2.2.

1.4 Motivations

We shall look at some motivations of our problem. First, we will see that finding an embedding for a graph was previously studied by Roberts, called the *indifference graph*. This implies that the naive case, *i.e.*, the binary Robinson matrices, always has a uniform embedding. Second, we will present an example that a Robinson matrix does not have uniform embeddings, with respect to any threshold vector.

It is known in [13] that a binary symmetric matrix A, where its diagonal entries are all 1's, is a Robinson matrix if and only if $A - I_n$ is the adjacency matrix of a proper interval graph. In [21], Roberts shows that each class of proper interval graphs and unit interval graphs is equal to the class of indifference graphs. Discussed in [25], any Robinson matrix in $\mathcal{S}^n[k]$ can be seen as a set of k nested unit interval graphs. Let $A = (a_{i,j}) \in \mathcal{S}^n[k]$, denote $G^{(t)} = ([n], E_t)$ as the *t-th level graph of A* for $t \in [k]$, where the edge sets E_t are defined by $\{u, v\} \in E_t \iff a_{u,v} \geqslant t$. Let $R^{(t)}$ denote the adjacency matrix of $G^{(t)}$. Then

$$A = D_n + \sum_{t=1}^{k} R^{(t)}, \tag{1.4.1}$$

where D_n is a diagonal matrix. Moreover, A is a Robinson matrix if and only if $R^{(t)} + I_n$ is a Robinson matrix for all $t \in [k]$. Then, by Roberts' result, the graph $G^{(t)}$ of each $R^{(t)}$ is some indifference graph with some threshold distance d_t. Denote the associated embedding by Π_t. If $k = 1$, then $A = D_n + R^{(1)}$. Roberts' result applies immediately that indifference graph $G^{(1)}$, equipped with Π_1 and d_1, is a uniform embedding of A with respect to d_1.

In this light, the problem of finding a uniform embedding of A with respect to $\boldsymbol{d} = (d_t) \in \mathbb{D}_k$ of a Robinson matrix is equivalent to finding an embedding Π so that the k indifference graphs equipped with Π and d_t has $R^{(t)}$ as its adjacent matrix, *simultaneously*. Notice that not all Robinson matrices has uniform embeddings.

Example 1.3. Consider the following two matrices $A = (a_{i,j}), B = (b_{i,j})$,

$$(a_{i,j}) = \begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 2 & 2 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 1 & 2 & 2 \end{pmatrix} \end{array} \quad (b_{i,j}) = \begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 2 & 2 & 1 & 0 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 & 1 \\ 0 & 1 & 2 & 2 & 2 & 1 \\ 0 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 1 & 1 & 2 & 2 \end{pmatrix} \end{array}$$

In these two cases, $A \in \mathcal{S}^5[2]$ and $B \in \mathcal{S}^6[2]$. Matrix A has a uniform embedding Π with threshold vector $\boldsymbol{d} = (d_1, d_2)^\top = (8, 6)^\top$,

i	1	2	3	4	5
$\Pi(i)$	0	5	6.5	11.75	12.75

For matrix B, suppose there is a uniform embedding Π with respect to some $d_1 > d_2 > 0$. Observe the following two inequalities are both implied by the constraints:

$$\begin{cases} \Pi(4) - \Pi(1) > d_1, \\ \Pi(6) - \Pi(4) > d_2 \end{cases} \Rightarrow d_1 + d_2 < \Pi(6) - \Pi(1);$$

$$\begin{cases} \Pi(2) - \Pi(1) \leqslant d_2, \\ \Pi(6) - \Pi(2) \leqslant d_1 \end{cases} \Rightarrow \Pi(6) - \Pi(1) \leqslant d_1 + d_2.$$

Combining the inequalities $d_1 + d_2 < \Pi(6) - \Pi(1) \leqslant d_1 + d_2$ results in a contradiction. Thus we conclude that matrix B does not have a uniform embedding.

Based on this example, we ask this question: **what are the characteristics of a Robinson matrix that has a uniform embedding?**

1.5 Main Results

In Chapter 2, we discuss the naïve case of $\mathcal{S}^n[k]$ when $k = 1$, *i.e.*, binary Robinson matrices. We will describe, by Roberts' result in [21], that any unit interval graph has a representation as an indifference graph, such indifference graph can be seen as a uniform embedding. Therefore, the binary Robinson matrices always have a uniform embedding. The recognition of the unit interval graph is studied extensively; we discuss one such strategy by scaling and translating the endpoints of the intervals, and discuss why this method does not apply to our thesis.

In Chapter 3, we present our solution to the problem: a sufficient and necessary condition so that a Robinson matrix has uniform embedding; we provide a constructive proof to find a uniform embedding if the condition is satisfied. The condition associates the upper/lower bound from the Robinson matrix (Definition 1.2) with walks in its graph (as in Example 1.3): we call them upper- and lower-bound-walks and they can be defined independently from the uniform embedding. The walks concatenate the edges that share the same endpoints, and thus, by combining the bounds obtained from the two edges, another upper/lower bound is implied by Definition 1.2. We consider all of the implied upper and lower bounds from Definition 1.2 of a Robinson matrix, and form them into a system of inequalities where the constraints are linear equations with the threshold distances as the variables. Then, we formulate our main theorem: the Robinson matrix has a uniform embedding if and only if the system of inequalities formed by the implied bounds has a solution. We also discuss that only the upper- and lower-bound-paths (*i.e.*, bound-walks that contain no repeating vertices) contribute to this inequality system. We show that the forward implication is trivial; we prove the converse implication by constructing a uniform embedding with a given threshold vector and the sets of all implied upper and lower bounds.

In Chapter 4, we discuss the complete procedure of computing a uniform embedding of a Robinson matrix: the computation in the proof of the converse implication requires the sets of upper and lower bounds and an appropriate threshold vector. We propose a Floyd-Warshall-like algorithm, "Bound-Generation" algorithm, to compute the upper- and lower-bound-paths in a Robinson matrix. Since the definition of upper- and lower-bound-walks are different from the definitions of walks and paths in conventional graph theory, we give the proof of correctness of our algorithm. We reformulate the inequality system in Chapter 3 in terms of upper- and lower-bound-walks so that the inequalities can be solved as a linear program and produce a solution $\boldsymbol{d} \in \mathbb{D}_k$.

We also consider optimizing the performance of the Bound-Generation algorithm by defining a partial order on the bounds, and we analyze the complexity of the procedure. We denote M as the size of minimal and maximal elements of upper and lower bounds under this partial order. Then we give a complexity analysis of the procedure in terms of M and n, where n is the size of the matrix.

Finally, we discuss the case of $k = 2$ and propose a combinatorial algorithm that computes a uniform embedding, *i.e.*, we find a threshold vector without using a linear program. Moreover, with the partial order, we employ Dilworth's theorem and rewrite the complexity analysis in terms of only n.

Chapter 2

Related Work

We devote this chapter to a review of work on Robinson matrices. We consider a Robinson matrix in our work and determine whether this matrix has a uniform embedding. Closely related to our work is the seriation problem. Section 2.1 introduces some work that connects Robinson matrices with graph theory. Among the results, the unit interval graphs associate with Robinson matrices closely. We discuss some properties of unit interval graphs, whose adjacency matrices with diagonal entries filled with 1s are binary Robinson matrices. Then, we connect the Robinson matrices as a stack of nested unit interval graphs. We introduce an algorithm that transform any proper interval graph into a unit interval graph; or equivalently, find a uniform embedding of any binary Robinson matrix. In Section 2.2, we then talk about a short history of the seriation problem and several works that recognize Robinson matrices. In Section 2.3, we discuss uniform linear embeddings of diagonally increasing graphons. *Graphons* are functions that can be seen as a generalization of matrices. Diagonally increasing graphons satisfy a property similar to property (1.2.1), which defines Robinson matrices. Uniform linear embeddings of graphons are similar to uniform embeddings of matrices. We discuss how the results (on graphons) relate to our results for matrices in Section 2.3.

2.1 Robinson matrix and Unit Interval Graph

In this section, we review some studies on the Robinson matrix. We look at some works that associate Robinson matrices with graph theory; our work builds up an intuition based on the graphs of Robinson matrices. We list some characterizations of the binary Robinson matrices provided in different works from [10], [21], [17], and [22]. We also discuss one construction of a unit interval graph from proper interval graphs by Bogart and West ([2]), which is a naïve case of finding a uniform embedding on binary Robinson matrices.

2.1.1 Graph Theory and Robinson Matrix

Flinders Petrie, as mentioned, invented the method of sequence dating. His method represents the objects and features by a matrix, where each row corresponds to an object and each column corresponds to a feature. The matrix with the arrangement on objects that reveals a solution to the underlying chronological order is called a *Petrie matrix*. To define the Petrie matrix, we need first to define the *Consecutive Ones Property (C1P)*. Consider a binary matrix $(a_{i,j})$. If there is a permutation τ so that the entries $a_{i,\tau(j)} = 1$ on rows are consecutive, this matrix is said to have the *C1P on rows*; similarly, $(a_{i,j})$ has *C1P on columns* if there is permutation τ so that entries $a_{\tau(i),j} = 1$ on columns are consecutive. A binary matrix has C1P if it has C1P on both rows and columns. A *Petrie matrix* is a vertex-clique incidence matrix that has C1P without permuting rows and columns. That is, the permutation τ is the identity permutation.

Kendall ([13]) showed that for any vertex-clique matrix P, P is a Petrie matrix if and only if $P^\top P$ is a Robinson matrix. With this equivalence, the problem of finding whether or not a vertex-clique matrix P has C1P reduces to finding whether there is a permutation matrix T, so that $(TP)(TP)^\top$ is a Robinson matrix.

Further, Roberts' characterization ([21]) also shows that a Robinson matrix corresponds to the adjacency matrix of a unit interval graph, *i.e.*, A is a Robinson matrix if and only if A is an generalized adjacency matrix of a unit interval graph: recall that a symmetric matrix $(a_{i,j})$ is a generalized adjacency matrix of $G = ([n], E)$ if $a_{i,j} \neq 0 \iff \{i, j\} \in E$. Thus, we may use graphs to describe the Robinson matrices in the latter content. In the following, we list some characteristics on the proper interval graph; and since the unit interval graphs are equivalent to proper interval graphs, these properties applies to Robinson matrices as well.

Theorem 2.1 ([10, 21, 17, 22])**.** *For graph $G = ([n], E)$, the following are equivalent:*

1. *G is a proper interval graph.*
2. *[10] The clique-vertex incidence matrix of G has the C1P.*
3. *[10] (Clique condition) There is a permutation $\tau \in \mathcal{P}_n$ such that the vertices contained in any maximal clique of G are consecutive with respect to τ.*
4. *[21] G is a unit interval graph.*
5. *[21] G is a indifference graph with some embedding Π and threshold distance d.*
6. *[21] G is a $K_{1,3}$-free interval graph (i.e., there is no $K_{1,3}$ as induced subgraph of G).*

7. *[17] (3-vertex condition) There is a permutation $\tau \in \mathcal{P}_n$ such that for all $x, y, z \in [n]$,*

$$\tau(x) < \tau(y) < \tau(z), \{x, z\} \in E \Rightarrow \{x, y\}, \{y, z\} \in E.$$

8. *[22] (Neighbourhood condition) There is a permutation $\tau \in \mathcal{P}_n$ such that for any $x \in [n]$ the vertices in $N[x]$ are consecutive with respect to τ.*

Although Gardi ([11]) proved the equivalence by Petrie matrices and the proper interval graphs (*i.e.*, (1) $\Rightarrow$ (2) $\Rightarrow$ (4) $\Rightarrow$ (1)) before examining these works, we take a detour to the more extensively studied topic, Robinson matrices.

2.1.2 Proper = Unit

In this subsection, we consider finding uniform embedding of a binary Robinson matrix. By Theorem 2.1, the problem of finding a uniform embedding of a binary Robinson matrix is equivalent to finding whether the graph of a matrix is a proper interval graph, *i.e.*, a graph is an indifference graph if and only if it is a unit interval graph, if and only if it is a proper interval graph. Booth and Lueker in [3] first invent one recognition algorithm for proper interval graphs, and later, more algorithms are proposed, such as [17, 12, 6]. We find a proof of the equivalence between proper interval graphs and unit interval graphs by [2] that constructs a unit interval graph from a proper interval graph. We will see how this graph problem can be seen as a special case of the uniform embedding problem.

Let $A = (a_{i,j}) \in \mathcal{S}^n[k]$ with $k = 1$, which means A is a binary Robinson matrix. Then, $A - I$ is the adjacency matrix of a unit interval graph as in Item 4 of Theorem 2.1. Let $G = (V, E)$, where $V = \{I_v = [a_v, b_v] : v \in [n]\}$ and $b_v - a_v = 1$ for all $v \in [n]$, be the unit interval graph representation of A. Take the middle point and denote by $\Pi(v) = (a_v + b_v)/2$, then $|\Pi(v) - \Pi(u)| \leqslant 1 \iff I_u \cap I_v \neq \emptyset$. Then, Π is a uniform embedding of Robinson matrix A with respect to 1. Notice that if we scale the embedding Π by any $d \in \mathbb{R}$, then $d\Pi$, defined as $(d\Pi)(v) = d \cdot \Pi(v)$, is a uniform embedding of A with respect to d. This concludes that any binary Robinson matrix has a uniform embedding with respect to any d. Thus, if the graph representation of a binary matrix is a unit interval graph, then the corresponding binary Robinson matrix has a uniform embedding.

Then, by Theorem 2.1, a graph is a proper interval graph if and only if it is a unit interval graph: Consider a unit interval graph and its intervals, if we take the middle point of each interval, then the two intervals are adjacent to each other if and only if the two middle points are within the unit length. Therefore, define a map that maps n vertices, $[n]$, to the middle points, such embedding is a uniform embedding of the corresponding Robinson matrix. Thus, a binary Robinson matrix is an generalized adjacency matrix of a proper interval graph if and only if it has a uniform embedding. As we mentioned, recognizing proper interval graphs are extensively studied. If we have an algorithm that transforms any proper interval graph to a unit interval graph, then we find a uniform embedding for the corresponding binary Robinson matrix. In [2], Bogart and West provide such an algorithm.

We will now describe the algorithm given in [2]. For a proper interval graph, the fact that no interval is completely contained in another implies that the intervals can be ordered such that $V = \{I_v = [a_v, b_v] : v \in [n]\}$ so that $\{a_v\}, \{b_v\}$ are both strictly increasing, *i.e.*, $a_v < a_{v+1}$ for all $v \in [n-1]$. The algorithm is intuitive. Iteratively, the algorithm adjusts the length of each interval $I_v = [a_v, b_v]$, for $v = 1, \dots, n$, to the unit length. If the length of I_v is less than the unit length, then translate the endpoints (both a_u's and b_u's) on the right of b_v to farther right; if the length of I_v is greater than the unit length, then scale all endpoints on the left of b_v, whose interval are not adjusted yet, proportionally so that all the endpoints remains the same order and is on the left of $a_v + 1$. Then, set new b_v to $a_v + 1$, which finish adjusting I_v. Precisely, the algorithm are defined as the following. Inductively, for all $v \in [n]$, suppose that all intervals I_u for $u < v$ are adjusted to unit length. Consider the following operation that adjusts interval I_v, for $v \in [n]$.

1. Set $\alpha = \max\{a_v\} \cup \{b_u : b_u \in I_v\}$;

2. Scale proportionally or translate all b_u that are between b_v and α. Precisely, if $\alpha < b_v$, then scale $b_u \in (\alpha, b_v]$ to $(\alpha, a_v + 1]$, with function $f(x) = \frac{a_v+1-\alpha}{b_v-\alpha}x + \frac{b_v-a_v-1}{b_v-\alpha}$; If $b_v \leqslant \alpha$, then translate $b_w \in [b_v, \infty)$ by $\alpha - b_v$ for all $v < w$. Finally, set new $b_v = a_v + 1$.

By the inductive hypothesis, all intervals before v are adjusted to the unit length. And since we assumed that $\{a_v\}, \{b_v\}$ are strictly increasing, then we have that $\alpha < \min\{a_v + 1, b_v\}$, *i.e.*, for $u < v$, $a_u < a_v$ and $b_u = a_u + 1$, so $b_u = a_u + 1 < a_v + 1$, and since the graph is proper, there are no $u < v$ such that $b_u \geqslant b_v$. By the choice of α, the endpoints b_u for $u < v$ are not involved in the adjustment of I_v. Finally, the scaling and the translation does not change the order of a_w, b_w for all $v < w$, therefore the graph remains the same. Therefore, the above procedure produces an interval graph with the same adjacency matrix while all the intervals are of unit length.

In our work, for general Robinson matrices (*i.e.*, in $\mathcal{S}^n[k]$), however, we cannot adapt the similar scaling/translating strategy. Intuitively, we can rewrite a Robinson matrix into a family of binary Robinson matrices as in Equation (1.4.1). However, notice that Definition 1.2 consist of an existence of a vector $\boldsymbol{d} = (d_i)$. When $k = 1$, $\boldsymbol{d} = (d_1)$ can be arbitrary without loss of generality. It is not shown, when $k > 1$, that there exists a vector (d_i) that satisfies the condition. Thus, we cannot scale and translate according to d_i's. Alternatively, let a Robinson matrix be decomposed into a family of nested proper interval graphs according to Equation (1.4.1), then transform into a set of the unit interval graphs. However, defining a uniform embedding from a unit interval graph is to take the middle point of each interval. Let $I_v^{(t)}$ to be the interval corresponds to vertex v in level graph $G^{(t)}$. Then, it requires the algorithm to transform the middle points of $I_v^{(t)}$, for all $t \in [k]$, to the same point in order to define the uniform embedding with this strategy.

2.2 The Seriation Problem

In this section, we discuss the recognition of Robinson matrices. A symmetric matrix A is a *Robinsonian matrix* if there exists a permutation matrix T so that $TAT^\top$ is a Robinson matrix. The permutation τ corresponding to T is called a *Robinson order*. This setting is natural: recall the archaeology example, such that given a set of objects and their pair-wise similarities as the entries, in order to find their sequence dating, we need to find a permutation matrix so that it permutes columns and rows to make it a Robinson matrix.

2.2.1 Seriation Problem

Given the similarities, finding the ordering so that the similarities have the "closer is more similar" is known as the *seriation problem*. The seriation problem is originally invented as an archaeological word to describe a method of *relative dating*: when *the methods of absolute dating* (such as stratigraphy or radio carboning that directly determines the age of the objects) cannot be applied to the newfound artifacts or relics, archaeologists date the artifacts by their similarities to the others.

The bridge between math and archaeology is introduced after a series of works by Petrie ([19]), Robinson ([23]), and Kendall ([13]). Flinders Petrie is commonly recognized as one of the most important archaeologists who connects archaeology and mathematics in his work *Sequences in prehistoric remains* ([19]). Petrie invented sequence dating, in which it does not precisely provide the date (year/era) of the manufacture date of the set of objects but provides a linear order on them. The Petrie matrix is a common way of solving the seriation problem nowadays. In archaeology, a Petrie matrix is a $n \times m$ matrix, where n is the number of items and m is the number of identified *features*. Entry $a_{i,e}$ is set to 1 if object i has the feature e. Matrices in this form are nowadays known as the vertex-clique incidence matrices.

W. Robinson put down another milestone on math and archaeology. Consider a set of relics which are ordered as $[n] = \{1, 2, \ldots, n\}$ according to their actual underlying chronological sequence. Denote $a_{i,j}$ as the similarity between relics i and j, and notice that $a_{i,j} = a_{j,i}$ for any two relics. If the ordering $[n]$ on the objects indeed corresponds to their chronological order, then the similarities satisfy Equation (1.2.1), thus Robinson matrix is named after Robinson.

Readers may refer to the review [16] by Liiv for more background on archaeology and seriation.

2.2.2 Recognizing Robinson Matrix

Suppose a matrix that consists of similarity measurements, but the measures are obtained by "guessing", then it is natural to ask whether this "guessing" is accurate or not. It is also a presumption of our work, such that we assume the given matrix is a Robinson matrix. In this subsection, we look at some works done on recognizing whether a matrix is Robinsonian matrix, such as [3, 1, 6]. Readers may consult [25] by Seminaroti who summarized the algorithms proposed during the past half a century, in Table 3.6 and the corresponding section.

PQ-tree

Notice that for every Robinson matrix, there are at least two Robinson orders: one Robinson order and its reversal; that is, the Robinson matrix itself and the Robinson matrix obtained by permuting with permutation $(n, n-1, \ldots, 2, 1)$. For example,

$$\begin{pmatrix} 2 & 2 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 1 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 & 2 & 1 & 1 & 0 \\ 2 & 2 & 2 & 1 & 0 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Moreover, if two rows are identical to each other rows, then we may reorder arbitrarily and the resulting matrix is still a Robinson matrix. Booth and Lueker ([3]) designed a data structure, PQ-tree, to store a family of permutations on the set $[n]$. A PQ-tree is a tree where each element of $[n]$ is a leaf node, and each non-leaf node is either a P-node or a Q-node, each node has an ordering on its child nodes. The definition of a PQ-tree is as the following. Based on one Robinson order, another permutation is also a Robinson order if it can be generated by arbitrarily reordering the child nodes of P-nodes while preserving the order of the child nodes of Q-nodes up to its reversal. In [20, 18], two algorithms are proposed to recognize a Robinsonian matrix using PQ-tree.

A Spectral Algorithm

In [1], Atkins *et al.* proposed a spectral algorithm that recognizes the Robinsonian matrix based on the Fielder vector of its Laplacian matrix. The *Laplacian matrix* L_A of a symmetric matrix $A = (a_{i,j})$ is the matrix $L_A = D_A - A$ where $D_A = (d_{i,j})$ is a diagonal matrix where $d_{i,i} = \sum_{j \in [n]} a_{i,j}$. The *Fiedler value* is the second smallest eigenvalue, and an eigenvector corresponds to the Fiedler value is a *Fiedler vector*. For simplicity to explain the intuition, we assume the given matrix is already a Robinsonian matrix, is irreducible, and that contains no repeating rows. In [1], the algorithm solves all the cases and return FALSE if the matrix is not Robinsonian; we omit the complete procedure due to the relevancy of the content.

Atkins *et al.* show in [1] that, if a Robinson matrix A contains no repeating rows, then any Fiedler vector of its Laplacian matrix is strictly increasing or decreasing (*i.e.*, let $\boldsymbol{x} = (x_i)$ be a Fiedler vector, $x_i > x_{i+1}$ for all $1 \leqslant i < n$ or $x_i < x_{i+1}$ for all $1 \leqslant i < n$).

Let T be a permutation matrix with appropriate dimension. The algorithm uses the following linear algebra property. Notice that any two *similar matrices* share the same characteristic polynomial; or equivalently, if matrices A and B has $B = TAT^\top$, then the eigenvalues of A and B are the same. Let λ be an eigenvalue of A and let $\boldsymbol{x}$ the corresponding eigenvector, then

$$(TAT^\top)T\boldsymbol{x} = TA\boldsymbol{x} = T(\lambda\boldsymbol{x}) = \lambda T\boldsymbol{x};$$

that is, $T\boldsymbol{x}$ is the eigenvector corresponds to λ as an eigenvalue of $TAT^\top$. The algorithm in [1] uses this property, where the subroutine is to find a permutation matrix T that sorts a Fiedler vector of A in an increasing order, $T\boldsymbol{x}$. Then A is a

Robinsonian matrix if and only if $TAT^\top$ is a Robinson matrix.

A Combinatorial Algorithm

Seriation problem for binary Robinson matrices is essentially a graph theory problem. We mentioned that Roberts characterized that if a Robinson matrix is a generalized adjacency matrix of a unit interval graph, then a list of properties in Theorem 2.1 applies to Robinson matrices as well. Also recall that Equation (1.4.1) rewrites a Robinson matrix in $\mathcal{S}^n[k]$ as the adjacency matrix of a set of nested unit interval grpahs. Therefore, finding an Robinson order of a matrix is equivalent to finding a permutation τ so that each level graphs of the matrix satisfies the neighbour condition (Item 8 in Theorem 2.1).

A series of works done by Habib ([12]), Corneil ([7, 6]), and Laurent and Seminaroti ([14, 15]) provides several combinatorial algorithms to recognize the Robinson matrix. More precisely, to find a Robinson order of a Robinsonian matrix is to find a permutation τ such that, for each level graph of A, τ satisfies the neighbour condition (Item 8) in Theorem 2.1. In [6], Corneil proposed a *3-sweep algorithm* that recognizes the unit interval graphs base on the fact that unit interval graph has a perfect elimination ordering. A *perfect elimination ordering* is an ordering, say $[n]$, so that for each $i \in [n], N(i) \cap [i]$ forms a clique. In [24], Rose and Tarjan proposed a variation of Breath-First Search algorithm (BFS algorithm), *LexBFS* algorithm, that breaks ties in BFS algorithm according to a given order on the vertices in a graph. Habib ([12]) applied the LexBFS algorithm to recognize chordal graphs by recognizing its simplicial vertex. Then, combining the fact that each unit interval graph contains at least two simplicial vertices, Corneil proposed the 3-sweep algorithm that applies Lex-BFS to a unit interval graph in [6]. Further, a graph is a unit interval graph if and only if the ordering returned by the 3-sweep algorithm satisfies the neighbour condition. Laurent and Seminaroti [14] proposed the algorithm, based on the 3-sweep algorithm and that the Robinson matrix is a set of nested unit interval graphs, "refines" the permutation so that it satisfies the neighbour condition for all level graphs $G^{(t)}$.

2.2.3 Time Complexity

In this thesis, we focus on the combinatorial properties of the algorithms; therefore, we denote complexity of computing the eigenvalues as a variable, $T(n)$, since the algorithm involves some numerical analysis: it depends on which package does the implementation use. All the above algorithms are in a polynomial time complexity:

In [1], the algorithm by Atkins *et al.*runs in $O(n(T(n) + n \log n))$; the algorithm by Laurent and Seminaroti ([14]) runs in $k(m + n)$, where m is the number of positive entries in the matrix and k is the maximum value in the entries (*i.e.*, $\mathcal{S}^n[k]$). In [25], Seminaroti gives a survey about the algorithms on recognizing Robinson matrices, and provides their novel algorithm base on the *Similarity-First Search*, which runs in $O(n + m \log m)$. In this thesis, we will see that the procedure that determines and computes a uniform embedding has a higher complexity than recognizing a Robinson matrix. Therefore, in this thesis, we assume the matrices are given in the form of Robinson matrix.

2.3 Uniform Linear Embeddings of Graphons

In this section, we discuss another important motivation of this thesis from [5] by Chuangpishit, Ghandehari, and Janssen. Namely, these authors in [4] consider a problem very similar to the one considered in this thesis, namely the problem of finding a uniform embedding of a graphon rather than a matrix.

Let $\mathcal{W}_0$ denote the set of all symmetric measurable functions $w : [0, 1]^2 \to [0, 1]$. A function $w \in \mathcal{W}_0$ is called a *graphon.* A graphon w is *diagonally increasing* if, for all $x, y, z \in [0, 1]$,

$$\begin{aligned} w(x, y) &\leqslant w(x, z) \quad \text{if } x \leqslant y \leqslant z, \\ w(y, x) &\geqslant w(z, x) \quad \text{if } y \leqslant z \leqslant x. \end{aligned}$$

The analysis in [5] assumed the graphons to be finitely valued, that is, let the range of function w be $\text{range}(w) = \{\alpha_1, \ldots, \alpha_k\}$, where $\alpha_1 > \alpha_2 > \cdots > \alpha_k$. A graphon w has a *uniform linear embedding* if there is an embedding $\pi : [0, 1] \to [0, 1]$ and real numbers $0 < d_1 < \cdots < d_k$ so that, for all $x, y \in [0, 1]$,

$$\begin{cases} w(x, y) = \alpha_i \iff |\pi(x) - \pi(y)| \leqslant d_k \\ w(x, y) = \alpha_i \iff d_i < |\pi(x) - \pi(y)| \leqslant d_{i+1} \text{ and } 1 < i < k \\ w(x, y) = \alpha_i \iff d_1 < |\pi(x) - \pi(y)| \end{cases}$$

Notice that the definition of a diagonally increasing graphon is similar to the definition of a Robinson matrix (*i.e.*, 1.2.1), and the definition of the uniform linear embedding is similar to uniform embedding. In the context of this thesis, a Robinson matrix is a special case of graphons. Divide $[0, 1]$ into n intervals, $I_i = (b_i, b_{i+1})$, where $b_1 = 0$ and $b_{n+1} = 1$. Define graphon $w(x, y) = a_{u,v} \iff x \in I_u, y \in I_v$. Intuitively, obtaining a graphon from a Robinson matrix is to divide the unit square $[0, 1]^2$ into $n \times n$ grids and assign value to each u, v-th grid by the entry value $a_{u,v}$.

We could also notice that, for any graphon w that is finitely valued (*i.e.*, as defined above), w can be determined by sets of functions as the following. Define $l_i, r_i : [0,1] \to [0,1]$ as

$$\begin{aligned} l_i(x) &= \inf\{y \in [0,1] : w(x,y) \geqslant \alpha_i\} \quad \text{and} \\ r_i(x) &= \sup\{y \in [0,1] : w(x,y) \geqslant \alpha_i\}. \end{aligned}$$

In other words, l_i's and r_i's, for $i \in [k]$, indicates the boundary between the regions where $w(x,y) \geqslant \alpha_i$ and $w(x,y) < \alpha_i$.

The approach in [5] concatenates l_i's and r_i's with appropriate domains to partition $[0,1]$ into intervals; and for each interval, define a uniform linear embedding of the graphon. This approach is technical but based on the functions l_i and r_i, and gives a sufficient and necessary condition of a diagonally increasing function that has a uniform linear embedding. However, we may not adapt their result to this thesis. In their work, they assumed that set l_i, r_i are strictly increasing; but, for Robinson matrices, the boundary functions are the boundaries along the grids, which are defined naturally to be step functions, and the step functions are not strictly increasing. As for when l_i, r_i that are not strictly increasing, the result does not apply to our case.

Chapter 3

Uniform Embedding

In this chapter, we aim to prove the main result of this thesis, Theorem 3.18. Recall the definition of uniform embedding, Definition 1.2, stated again here.

> Given a matrix $A \in \mathcal{S}^n[k]$ *and a threshold vector* $\boldsymbol{d} \in \mathbb{D}_k$, a map $\Pi : [n] \to \mathbb{R}$ is a *uniform embedding of* A *with respect to* $\boldsymbol{d}$ if, for each pair $u, v \in [n]$ (1.3.1) is satisfied:
>
> $$a_{u,v} = t \iff d_{t+1} < |\Pi(v) - \Pi(u)| \leqslant d_t \quad \text{for } t \in \{0, \dots, k\},$$
>
> where we define $d_{k+1} = -\infty$ and $d_0 = \infty$, so that the lower bound for $a_{u,v} = k$ and the upper bound for $a_{u,v} = 0$ are trivially satisfied.

We say that matrix A *has a uniform embedding* if there exists some $\boldsymbol{d} \in \mathbb{D}_k$ so that A has a uniform embedding with respect to $\boldsymbol{d}$.

We will show that a uniform embedding with respect to $\boldsymbol{d}$ exists if and only if all the inequalities implied by Definition 1.2 do not contradict each other, as in Example 1.3: this is immediate for the implication, yet the converse requires more tools to see. In Section 3.1, we study the form of the embedding Π of a uniform embedding and show that, if a matrix has a uniform embedding, then it also has one that is strictly increasing. Further, we show that, if a uniform embedding exists, then there is a uniform embedding that also satisfies (1.3.1) but where both upper and lower bound are strict inequalities. Section 3.2 defines the bounds in Example 1.3 in terms of k-dimensional integer vectors and independent of the threshold vector $\mathbb{D}_k$; we define these bounds independent from the threshold vector $\boldsymbol{d}$, but associate the bounds with the edges in the graph of a Robinson matrix. We obtain walks and paths from concatenating edges, and depending on the upper/lower bounds they induce, we call them upper- and lower-bound-walks (or bound-paths). We then observe that the upper- and lower-bound-paths (*i.e.*, walks that contain no repeating vertices) are sufficient to determine the existence of a uniform embedding of a given Robinson matrix A: We express this condition in terms of a system of inequalities that involves only the variable $d_1, \dots, d_k$. This system has a solution if and only if the matrix has uniform embedding.

3.1 Strict Monotonicity of Uniform Embedding

In this section, we look at the form of a uniform embedding of a Robinson matrix. This is expressed in the following theorem.

Theorem 3.1. *Let $A \in \mathcal{S}^n[k]$. If A has a uniform embedding, then there exists $\boldsymbol{d} \in \mathbb{D}_k$ and a uniform embedding Π of A with respect to $\boldsymbol{d}$ which is strictly increasing, and which is such that the inequalities in (1.3.1) are all strict. That is, for all pairs of $u, v \in [n]$, $u < v$,*

$$a_{u,v} = t \iff d_{t+1} < \Pi(v) - \Pi(u) < d_t, \tag{3.1.1}$$

where $d_{k+1} = 0$ and $d_0 = \infty$.

We assume the Robinson matrices in this section all have a uniform embedding (so we do not need to repeat "if the matrix has a uniform embedding" every time). Suppose Π is a uniform embedding of Robinson matrix A, we call $\Pi(u)$ as the *uniform embedding of vertex u* or the *uniform embedding of row u*. Notice, if the embedding of two repeating rows u and v is $\Pi(u) = \Pi(v)$, it does not violate inequalities (1.3.1). However, to be able to distinguish different vertices, it is natural to look for a uniform embedding that is injective. The strict lower bound of (3.1.1) implies that the uniform embedding has the nice property of being injective. The removal of the absolute signs together with the lower bound imply that a uniform embedding satisfying (3.1.1) is strictly increasing. Thus, when we compute a new distance between the embedding of vertices based on an old bound, for $u < v < w$, then we can write $\Pi(w) - \Pi(u) = \Pi(w) - \Pi(v) + \Pi(v) - \Pi(u)$ without the absolute value signs. Another rewrite is that the strict upper bound on $\Pi(v) - \Pi(u)$: this avoids the future proofs from dividing into cases such that $|\Pi(u) - \Pi(v)| = d_t$ or $|\Pi(u) - \Pi(v)| < d_t$, where $t = a_{u,v}$.

We break down the theorem into several steps. First, Lemma 3.2 proves that, if a strictly increasing uniform embedding satisfies Condition (1.3.1), then we can construct another uniform embedding with Condition (3.1.1). We then justify why we assume $\Pi(1) = 0$ without loss of generality in Lemma 3.3. Then, we show that the uniform embedding of a Robinson matrix with no repeating rows is always strictly increasing in Lemma 3.4. By combining Lemma 3.2 and Lemma 3.4, we prove Theorem 3.1 by showing that we can place the embedding of all repeating rows close enough, while the embedding remains strictly increasing.

Lemma 3.2. *Let Π be a uniform embedding of $A \in \mathcal{S}^n[k]$ with respect to $\boldsymbol{d} \in \mathbb{D}_k$ and suppose Π is strictly increasing. Then, there exists an increasing Π' where all inequalities are strict, i.e., satisfies Condition (3.1.1).*

Proof: Let Π be a uniform embedding of $A \in \mathcal{S}^n[k]$ with respect to $\boldsymbol{d} \in \mathbb{D}_k$ that is strictly increasing. Write $\boldsymbol{d} = (d_i)$. If $d_{t+1} < \Pi(v) - \Pi(u) < d_t$ where $t = a_{u,v}$ for all $u, v \in [n]$ with $u < v$, then $\Pi' = \Pi$ satisfies the statement of the lemma. Therefore, we assume there exists at least one pair $u, v \in [n]$ such that $\Pi(v) - \Pi(u) = d_t$, where $t = a_{u,v}$. Let $u \in [n]$ to be minimum vertex (index) such that there is $v \in [n]$ with $\Pi(v) - \Pi(u) = d_t$ where $t = a_{u,v}$. Let v be the minimum as well. Notice, for any $i, j \in [n]$ with $i < j$, $t = a_{i,j}$ implies that $d_{t+1} < \Pi(j) - \Pi(i)$, and thus $\Pi(j) > d_{t+1} + \Pi(i)$. Let $\epsilon = \min_{i,j \in [n], i<j}\{\Pi(j) - \Pi(i) - d_{t+1} : a_{i,j} = t\}$, and notice that $\epsilon > 0$. Then, define

$$\Pi_0(i) = \begin{cases} \Pi(i) & \text{for } i \leqslant u, \\ \Pi(i) - \frac{\epsilon}{2} & \text{for } i > u. \end{cases}$$

Such Π_0 is a uniform embedding of A with respect to $\boldsymbol{d}$:

1. For any pairs i, j both in $\{1, \ldots, u\}$ or both in $\{u+1, \ldots, n\}$, $\Pi(j) - \Pi(i) = \Pi_0(j) - \Pi_0(i)$;

2. For any $i \in \{1, \ldots, u\}, j \in \{u+1, \ldots, n\}$,

$$\Pi(j) - \Pi(i) - \epsilon < \Pi(j) - \frac{\epsilon}{2} - \Pi(i) = \Pi_0(j) - \Pi_0(i),$$

   where the inequality holds since $\epsilon > 0$ and the equality is given by the definition of Π_0.

Observe that $\Pi_0(v) - \Pi_0(u) = \Pi(v) - \frac{\epsilon}{2} - \Pi(u) = d_t - \frac{\epsilon}{2} < d_t$, so that pair u, v satisfies (3.1.1); and there is no new pairs $u', v' \in [n]$ so that $\Pi_0(v') - \Pi_0(u') = d_t$. Iteratively, obtain $\Pi_1, \Pi_2, \ldots, \Pi_r$ until all such u, v pairs, where $t = a_{u,v}$ and $\Pi(v) - \Pi(u) = d_t$, are adjusted to satisfy (3.1.1). Then, $\Pi' = \Pi_r$ is a uniform embedding of A with respect to $\boldsymbol{d}$ with all pairs of $u < v$, $a_{u,v} = t \iff d_{t+1} < \Pi(v) - \Pi(u) < d_t$ (*i.e.*, (3.1.1)). □

Lemma 3.3. *If Robinson matrix $A \in \mathcal{S}^n[k]$ has uniform embedding, then A has a uniform embedding Π' with $\Pi'(1) = 0$.*

Proof: Let Π be a uniform embedding of Robinson matrix A with respect to $\boldsymbol{d}$. Define Π' as $\Pi'(v) = \Pi(v) - \Pi(1)$ for $v \in [n]$, observe that Π' is also a uniform embedding of A with respect to $\boldsymbol{d}$ since the distance between $\Pi(v) - \Pi(u)$ is retained. Meanwhile, $\Pi'(1) = \Pi(1) - \Pi(1) = 0$, which was what we want. □

This lemma essentially allows us to simultaneously translate images of Π and set $\Pi(1) = 0$ without loss of generality.

Lemma 3.4. *Let $A \in \mathcal{S}^n[k]$ be a Robinson matrix with no repeating rows. Suppose Π is a uniform embedding of A with respect to $\boldsymbol{d} \in \mathbb{D}_k$, then Π is strictly monotone. If $\Pi(1) < \Pi(2)$, then Π is strictly increasing; and if $\Pi(1) > \Pi(2)$, then Π is strictly decreasing.*

Proof: Let $A \in \mathcal{S}^n[k]$ that contains no repeating rows. Suppose that Π is a uniform embedding of A with respect to $\boldsymbol{d} \in \mathbb{D}_k$. We will prove that Π is strictly increasing, that is, assume $\Pi(1) < \Pi(2)$, we will show $\Pi(i) < \Pi(i+1)$ for all $i < n$.

To prove this lemma, we break it down to two parts. First we prove that all vertices are embedded to distinct values. Second, we prove that the embedding Π is increasing. And thus, Π is strictly increasing.

We assumed that there are no repeating rows in the matrix, therefore, the embedding of the rows are distinct. To prove it, toward contradiction, suppose $\Pi(u) = \Pi(v)$, then

$$d_{t+1} < |\Pi(w) - \Pi(u)| = |\Pi(w) - \Pi(v)| \leqslant d_t$$

for all $w \in [n]$, and by Definition 1.2, $a_{u,w} = a_{v,w} = t$. This contradicts to the assumption that A contains no repeating rows.

We now assume $\Pi(1) \leqslant \Pi(2)$. We proceed an inductive proof such that Π restricted to $[v]$ is strictly increasing for each v from 2 to n. The base case, $\Pi(1) < \Pi(2)$, holds by the assumption and that $\Pi(1) \neq \Pi(2)$.

Inductively, suppose $v \geqslant 2$ and Π is strictly increasing restricted to $[v-1]$. By the inductive hypothesis, it suffices to prove that $\Pi(v-1) < \Pi(v)$ to show Π is strictly increasing restricted to $[v]$. Since there are no repeating vertices in A, let $w \in [n]$ be a vertex such that $a_{v-1,w} \neq a_{v,w}$.

1. Suppose $w < v$. By the definition of Robinson matrix, (1.2.1), $a_{v-1,w} > a_{v,w}$. Denote $t_1 = a_{v-1,w}, t_2 = a_{v,w}$ for some $t_1 > t_2$. By Definition 1.2, $d_{t_1} < d_{t_2}$ and

$$d_{t_1+1} < \Pi(v-1) - \Pi(w) \leqslant d_{t_1} \leqslant d_{t_2+1} < \Pi(v) - \Pi(w) \leqslant d_{t_2}. \tag{3.1.2}$$

   In (3.1.2), the sequence of strict inequalities on the left and the right is from Definition 1.2, the non-strict inequality in the middle is because $t_1 > t_2$ but $t_1 = t_2 + 1$ is possible. Rewrite (3.1.2) to obtain $\Pi(v-1) < \Pi(v)$.

2. Suppose w has $v < w$. First need to show $\Pi(v) \leqslant \Pi(w)$. Let $t_1 = a_{v-1,w}, t_2 = a_{v,w}$, where $t_1 < t_2$, by the definition of Robinson matrix, (1.2.1). Then, by Definition 1.2, $d_{t_1} > d_{t_2}$ and

$$d_{t_2+1} < \Pi(w) - \Pi(v) \leqslant d_{t_2} \leqslant d_{t_1+1} < \Pi(w) - \Pi(v-1) < d_{t_1}. \tag{3.1.3}$$

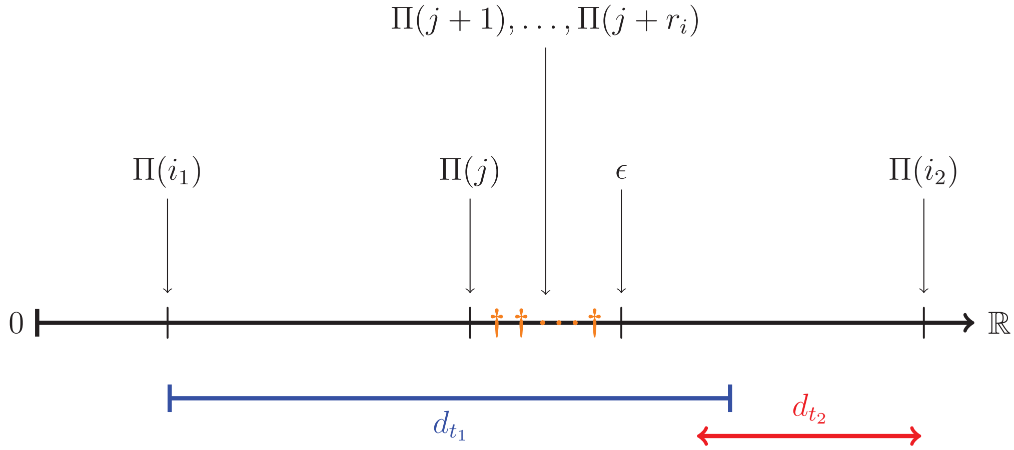

Figure 3.1: A sketch of proof of Theorem 3.1

Rewrite to obtain $\Pi(v-1) < \Pi(v)$.

This concludes $\Pi(v-1) < \Pi(v)$. Thus, if $\Pi(1) < \Pi(2)$, then Π is strictly increasing defined on any $[v]$ for v from 2 to n. If we assume $\Pi(1) > \Pi(2)$, then we can prove Π is strictly decreasing with the same logic. Thus, Π is strictly monotone defined on $[n]$. □

To complete the section, we prove Theorem 3.1. The proof is partially completed by Lemma 3.2 and Lemma 3.4 but missing the repeating rows. The proof of Theorem 3.1 is technical, so I lay down a proof sketch prior to the actual proof. Intuitively, the repeating rows are essentially a set of undistinguishable vertices, and therefore, we can map them to the same real number. However, for the statement of Theorem 3.1, we want to find a strictly increasing function, so we want to find a map that embeds the repeating rows to different values. We proved Lemma 3.2, so obtaining the strictly increasing mapping on the repeating rows is not hard to accomplish: consider a matrix A that has a uniform embedding, and let row $j, j+1$ be repeating rows in A. We first get a uniform embedding Π of the matrix obtained from A by removing all the repeating vertices except one. We can assume that Π is strictly increasing according to Lemma 3.4, this means it follows Condition (3.1.1). Then there always is space between $\Pi(j)$ and $\Pi(i) + d_t$ where $t = a_{i,j}$ (*i.e.*, $\Pi(j) < \Pi(i) + d_t$). Then, by the property of real numbers, we can fit any number of embedding on the interval $(\Pi(j), \Pi(j) + \epsilon)$ with ϵ small enough. Define $\Pi(j+1)$ by arbitrary number in the interval, $\Pi(j+1)$ satisfies Definition 1.2. Shown as in Figure 3.1.

Proof of Theorem 3.1: Suppose A has a uniform embedding with respect to $\boldsymbol{d} = (d_i)$. Let I be an index set that contains maximum number of non-repeating rows by their first appearance: for all $j \notin I$, there is $i \in I$ such that i, j are repeating rows and $i < j$.

Let Π' be a uniform embedding of A, then let $\Pi'' = \Pi'|_I$ defined on I where $\Pi''(v) = \Pi'(v)$ for all $v \in I$. Notice, the induced submatrix $A[I]$ is a Robinson matrix that contains no repeating rows and has a uniform embedding Π'' and Π'' is strictly increasing by Lemma 3.4.

Either Π'' satisfies (3.1.1) that all inequalities are strict, or there is $\Pi''(v) - \Pi''(u) = d_t$, where $t = a_{u,v}$; apply Lemma 3.2 to Π'', we obtain a uniform embedding $\Pi_0 : I \to \mathbb{R}$ that is a strictly increasing uniform embedding of $A[I]$ with respect to $\boldsymbol{d}$ that satisfies Condition (3.1.1) (*i.e.*, all inequalities are strict). For each $i \in I$, let

$$\begin{aligned} 2\epsilon_i = \min\{d_t - (\Pi_0(i) - \Pi_0(j)) : j < i, j \in I, a_{i,j} = t\} \cup \\ \{(\Pi_0(j) - \Pi_0(i)) - d_{t+1} : i < j, j \in I, a_{i,j} = t\}. \end{aligned} \tag{3.1.4}$$

Notice, Π_0 is a uniform embedding of $A[I]$ that is strictly increasing, *i.e.*, for all $i, j \in I$, if $i < j$, then $d_{t+1} < \Pi_0(j) - \Pi_0(i) < d_t$, and if $j < i$, then $d_{t+1} < \Pi_0(i) - \Pi_0(j) < d_t$: therefore $2\epsilon > 0$ (and thus $\epsilon > 0$).

(In other words, for each of $i \in I$, interval $(\Pi_0(i), \Pi_0(i) + 2\epsilon_i)$ defines the restricted space of placing the embedding of the repeating rows that are identical as row i. That is, for any row $i + k$ that is identical to row i, defining $\Pi_0(i + k) \geqslant \Pi_0(i) + 2\epsilon_i$ results in violating Π_0 being a uniform embedding; or equivalently, we may only define $\Pi_0(i + k) < \Pi_0(i) + 2\epsilon_i$ to obtain Π_0 as a uniform embedding. Moreover, the scalar "2" of 2ϵ can be any constant $c > 1$. That is, we will define all the embedding of repeating rows, that are identical to row i, strictly less than $\Pi_0(i) + c \cdot \epsilon$: precisely, define the repeating rows in interval $(\Pi_0(i), \Pi_0(i) + \epsilon_i]$, where the embedding of the repeating row with the largest index, $i + r_i$, is $\Pi_0(i + r_i) = \Pi_0(i) + \epsilon_i < \Pi_0(i) + c \cdot \epsilon$ for any $c > 1$.)

Let $i < j$ be a consecutive pair in I (*i.e.*, $i, j \in I$, there is no $k \in I$ such that $i < k < j$), denote r_i be the size of index set $\{i + 1, \ldots, j - 1\}$ where rows $i, i+1, \ldots, j-1$ are repeating rows that are identical as row i in A. Define $\Pi : [n] \to \mathbb{R}$ where $\Pi|_I = \Pi_0$, and for each $i \in I$, define Π for $j \notin I$ by

$$\Pi(i + k) = \Pi(i) + \frac{k}{r_i}\epsilon_i \quad \text{for } 1 \leqslant k \leqslant r_i.$$

Let $i \in I$ be arbitrary and we verify that Π is a uniform embedding of A. We divide into two cases, such that for all $j \in I, j \neq i$, either $i < j$ or $j < i$, we verify for all $k \in [r_i]$, $\Pi(j) - \Pi(i + k)$ or $\Pi(i + k) - \Pi(j)$ satisfies (3.1.1).

1. First, assume $j < i$. Let $1 \leqslant k \leqslant r_i$. By definition, $\Pi(i+k) - \Pi(j) = \Pi(i) - \Pi(j) + \frac{k}{r_i}\epsilon_i$. Notice the following inequalities,

$$\begin{aligned}\Pi(i) - \Pi(j) + \frac{k}{r_i}\epsilon_i &< \Pi(i) - \Pi(j) + \epsilon_i \\ &\leqslant \Pi(i) - \Pi(j) + d_t - (\Pi(i) - \Pi(j)) \\ &= d_t\end{aligned}$$

where $t = a_{i,j}$. This gives $\Pi(i+k) - \Pi(j) < d_t$. Recall that $a_{i+k,j} = t = a_{i,j}$ since $i, i+k$ are repeating rows, this gives:

$$a_{i+k,j} = a_{i,j} = t \iff d_{t+1} < \Pi(i) - \Pi(j) < \Pi(i+k) - \Pi(j) < d_t.$$

2. Next, assume $i < j$ and $1 \leqslant k \leqslant r_i$. By definition, $\Pi(j) - \Pi(i+k) = \Pi(j) - (\Pi(i) + \frac{k}{r_i}\epsilon_i)$ and observe that

$$\begin{aligned}\Pi(j) - \Pi(i) - \frac{k}{r_i}\epsilon_i &> \Pi(j) - \Pi(i) - \epsilon_i \\ &\geqslant \Pi(j) - \Pi(i) - (\Pi(j) - \Pi(i) - d_{t+1}) \\ &= d_{t+1}\end{aligned}$$

where $t = a_{i,j}$. This gives $\Pi(j) - \Pi(i) > d_{t+1}$. Recall $a_{i+k,j} = a_{i,j} = t$ and

$$a_{i+k,j} = t = a_{i,j} \iff d_{t+1} < \Pi(j) - \Pi(i) < \Pi(j) - \Pi(i+k) < d_t.$$

So the two cases establish that Π is a uniform embedding of A with respect to $\boldsymbol{d}$ and satisfies (3.1.1). $\square$

Note that the definition of d_{k+1} in Theorem 3.1 has changed from $-\infty$ to zero as in (3.1.1); this change enforces that Π is *strictly* increasing. In the later context, we will assume that any uniform embedding Π of matrix A with respect to any $\boldsymbol{d}$ is a map $\Pi : [n] \to \mathbb{R}$ which satisfies (3.1.1) (with the new definition of d_{k+1}). This will simplify the proofs and reduce the need to distinguish different cases. By Theorem 3.1, we can make this assumption without loss of generality.

3.2 Bounds, Walks, and Their Concatenation

Recall that matrix B in Example 1.3 does not have a uniform embedding since a contradictory bound is derived on $\Pi(6) - \Pi(1)$. Consider matrix B and its graph shown

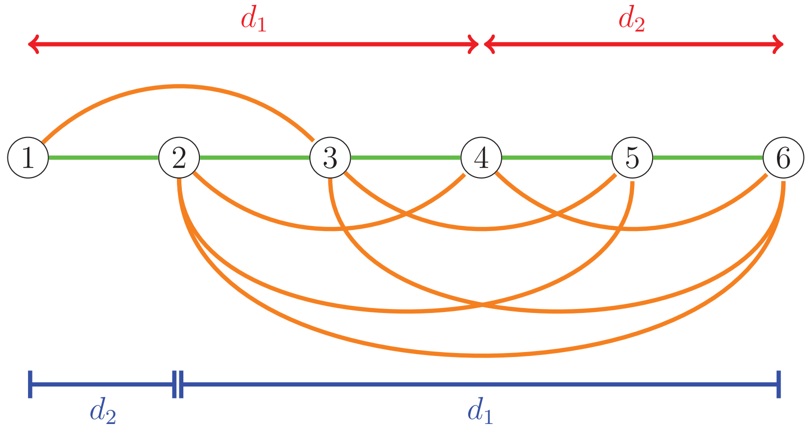

Figure 3.2: The graph of B in Example 1.3

in Figure 3.2, where orange edges represent the entries with value 1 and green edges represent the entries with value 2. In addition, let intervals with arrows $\overset{d_t}{\longleftrightarrow}$ indicate d_t is a lower bound on $\Pi(u)-\Pi(v)$ for any uniform embedding, *i.e.*, $d_t < \Pi(u)-\Pi(v)$; and the intervals with bars $\overset{d_t}{\vdash\!\!\!-\!\!\!\dashv}$ indicate d_t is an upper bound of $\Pi(u)-\Pi(v)$, *i.e.*, $\Pi(u)-\Pi(v) < d_t$. The contradiction on B also can be seen as a contradiction on a sequence of vertices $\langle 1, 4, 6, 2, 1\rangle$.

Definition 3.5. Let $A \in \mathcal{S}^n[k]$ and assume Π is a uniform embedding of A with respect to $\boldsymbol{d} \in \mathbb{D}_k$. Fix $u, v \in [n]$, $u < v$. A vector $\boldsymbol{b} \in \mathbb{Z}^k$ is an *upper bound on* (u, v) if the inequality $\Pi(v) - \Pi(u) < \boldsymbol{b}^\top \boldsymbol{d}$ is implied by the inequality system (3.1.1).

Similarly, the vector $\boldsymbol{b}$ is a *lower bound on* (u, v) if the inequality $\boldsymbol{b}^\top \boldsymbol{d} < \Pi(v) - \Pi(u)$ is implied by (3.1.1).

It follows directly from inequality system (3.1.1) that, for any matrix $A \in \mathcal{S}^n[k]$, and any pair $u, v \in [n]$, the all-zero vector $\mathbf{0}$ is a lower bound on (u, v) since we assumed all of the uniform embeddings are strictly increasing. We also extend the lower bound and upper bound on (u, u). Note that if $\boldsymbol{a}$ is a lower bound and $\boldsymbol{b}$ is an upper bound on (u, v) for any $u < v$, then $0 < (\boldsymbol{b} - \boldsymbol{a})^\top \boldsymbol{d}$ is implied. Therefore, a bound on (u, u) is an inequality involving only $d_1, d_2, \ldots, d_k$, and does not involve Π.

Remark 3.6. Given any uniform embedding Π of Robinson matrix, for any $u, v, w \in [n]$, the following equalities hold independently of $\boldsymbol{d} \in \mathbb{D}_k$.

$$\Pi(v) - \Pi(u) = \begin{cases} (\Pi(w) - \Pi(u)) - (\Pi(w) - \Pi(v)) & \text{if } u < v < w \qquad (3.2.1) \\ (\Pi(v) - \Pi(w)) - (\Pi(u) - \Pi(w)) & \text{if } w < u < v \qquad (3.2.2) \\ (\Pi(w) - \Pi(u)) + (\Pi(v) - \Pi(w)) & \text{if } u < w < v \qquad (3.2.3) \end{cases}$$

Lemma 3.7. *Let $A \in \mathcal{S}^n[k]$ and let $u, v, s \in [n]$ with $u < v$.*

1. If $u < s < v$, let $\boldsymbol{a}, \boldsymbol{a}'$ be lower bounds on $(u, s), (s, v)$, $\boldsymbol{b}, \boldsymbol{b}'$ be upper bounds on $(u, s), (s, v)$. Then $\boldsymbol{a} + \boldsymbol{a}'$ is a lower bound on (u, v) and $\boldsymbol{b} + \boldsymbol{b}'$ is an upper bound on (u, v).

2. If $s < u < v$, $\boldsymbol{a}, \boldsymbol{a}'$ be lower bounds on $(s, u), (s, v)$, $\boldsymbol{b}, \boldsymbol{b}'$ be upper bounds on $(s, u), (s, v)$. Then $\boldsymbol{a}' - \boldsymbol{b}$ is a lower bound on (u, v) and $\boldsymbol{b}' - \boldsymbol{a}$ is an upper bound on (u, v);

3. if $u < v < s$, $\boldsymbol{a}, \boldsymbol{a}'$ be lower bounds on $(u, s), (v, s)$, $\boldsymbol{b}, \boldsymbol{b}'$ be upper bounds on $(u, s), (v, s)$. Then $\boldsymbol{a} - \boldsymbol{b}'$ is a lower bound on (u, v) and $\boldsymbol{b} - \boldsymbol{a}'$ is an upper bound on (u, v).

Proof: Suppose A has uniform embedding Π with respect to any $\boldsymbol{d}$. Consider when $u < s < v$. Let $\boldsymbol{a}, \boldsymbol{a}'$ be lower bounds on $(u, s), (s, v)$, and let $\boldsymbol{b}, \boldsymbol{b}'$ be upper bounds on $(u, s), (s, v)$. The following inequalities are implied by (3.1.1),

$$\begin{aligned} \boldsymbol{a}^\top \boldsymbol{d} &< \Pi(s) - \Pi(u) < & \boldsymbol{b}^\top \boldsymbol{d} \\ \boldsymbol{a}'^\top \boldsymbol{d} &< \Pi(v) - \Pi(s) < & \boldsymbol{b}'^\top \boldsymbol{d} \end{aligned}$$

Apply the two sequences of inequality by Equation (3.2.3), then

$$(\boldsymbol{a} + \boldsymbol{a}')^\top \boldsymbol{d} < \Pi(v) - \Pi(u) < (\boldsymbol{b} + \boldsymbol{b}')^\top \boldsymbol{d}.$$

Other results can be obtained by the same logic. □

Let $\boldsymbol{\chi}_i \in \mathbb{Z}^k$ denote the unit vector with 1 at the ith position and zero otherwise and let $\mathbf{0}$ be the all-zero vector. Recall, we defined a symmetric matrix $A = (a_{i,j})$ is a generalized adjacency matrix of $G = ([n], E)$ if $\{i, j\} \in E$ if and only if $a_{i,j} \neq 0$.

Definition 3.8. Let $A = (a_{i,j}) \in \mathcal{S}^n[k]$ and let A be a generalized adjacency matrix of $G = ([n], E)$. Let $\{u, v\} \in E$, define *the upper bound induced by edge* $\{u, v\}$ as $\beta^+(\{u, v\}) = \boldsymbol{\chi}_t$ for $t \in [k]$ so that $a_{u,v} = t$, and $\beta^+(\{u, v\})$ is not defined if $a_{u,v} = 0$.

Similarly, define *the lower bound induced by edge* $\{u, v\}$ to be

$$\beta^-(\{u, v\}) = \begin{cases} \boldsymbol{\chi}_{t+1} & \text{if } a_{u,v} = t, t \leqslant k - 1; \\ \mathbf{0} & \text{if } a_{u,v} = k. \end{cases}$$

For $u, v \in [n]$ so that $\{u, v\} \notin E$, define $\{u, v\}$ to be a *null-edge*. In other word, there is a null-edge $\{u, v\}$ if $a_{u,v} = 0$. Define the *lower bound induced by null-edge* $\{u, v\}$ by $\beta^-(\{u, v\}) = \boldsymbol{\chi}_1$.

Lemma 3.9. *Let $A = (a_{i,j}) \in \mathcal{S}^n[k]$, let $u, v \in [n]$ and $u < v$. The upper bound induced by edge $\{u, v\}$, $\beta^+(\{u, v\})$, is an upper bound on (u, v); the lower bound induced by edge or null-edge $\{u, v\}$, $\beta^-(\{u, v\})$, is a lower bound on (u, v).*

Proof: The proof follows from Condition (3.1.1) immediately. For all uniform embedding Π of A with respect to any $\boldsymbol{d}$, for any $u, v \in [n]$, $u < v$, $\beta^+(\{u, v\}) = \boldsymbol{\chi}_t$ and $\beta^-(\{u, v\}) = \boldsymbol{\chi}_{t+1}$ or $\beta^-(\{u, v\}) = \mathbf{0}$ if $a_{u,v} = k$. Recall we assume without loss of generality that Π satisfies (3.1.1),

$$d_{t+1} = \boldsymbol{\chi}_{t+1}^\top \boldsymbol{d} < \Pi(v) - \Pi(u) < \boldsymbol{\chi}_t^\top \boldsymbol{d} = d_t;$$

or when $\beta^-(\{u, v\}) = \mathbf{0}$, the lower bound is 0; or when $a_{u,v} = 0$, $\Pi(v) - \Pi(u)$ is unbounded from above. Therefore, $\beta^+(\{u, v\})$ is an upper bound on (u, v) and $\beta^-(\{u, v\})$ is a lower bound on (u, v). □

In addition, we use bounds to rewrite (3.1.1) with bounds. For all $t \in [k-1]$, for all $u, v \in [n]$ where $u < v$,

$$a_{u,v} = t \iff \beta^-(\{u, v\})^\top \boldsymbol{d} < \Pi(v) - \Pi(u) < \beta^+(\{u, v\})^\top \boldsymbol{d}, \tag{3.2.4}$$

and for $t = 0$, $\beta^-(\{u, v\})^\top \boldsymbol{d} < \Pi(v) - \Pi(u)$.

Example 3.10. We use matrix $A = (a_{i,j})$ in Example 1.3,

$$(a_{i,j}) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 2 & 2 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 1 & 2 & 2 \end{pmatrix} \end{array},$$

as a generalized adjacency matrix of the graph in Figure 3.3. Coloured with the same rule as in Figure 3.2, the orange edges represent the entries with value 2, green edges are the entries with value 1, and pairs without an edge represent the entries with value 0, or say they are null-edges. Notice that upper bound $\beta^+(\{1, 5\})$ is not defined, but there are implied upper bounds: consider $\beta^+(\{1, 3\}) = \boldsymbol{\chi}_1$ and $\beta^+(\{3, 5\}) = \boldsymbol{\chi}_1$. For any uniform embedding Π of A with respect to any $\boldsymbol{d}$, Π satisfies inequalities (3.1.1) and all the implied inequalities, then $\Pi(5) - \Pi(1) = \Pi(3) - \Pi(1) + \Pi(5) - \Pi(3) < (\beta^+(\{1, 3\}) + \beta^+(\{3, 5\}))^\top \boldsymbol{d} = (\boldsymbol{\chi}_1 + \boldsymbol{\chi}_1)^\top \boldsymbol{d}$. Therefore,

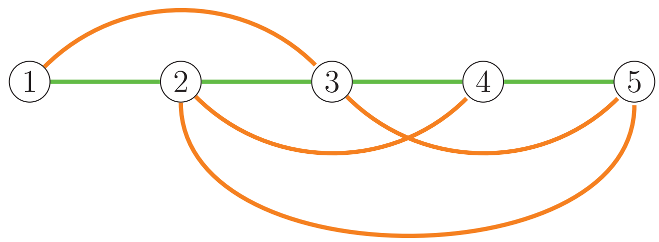

Figure 3.3: The graph of matrix A in Example 1.3

it is natural that $2\boldsymbol{\chi}_1 = (2,0)^\top$ is an upper bound on vertices $(1,5)$. Similarly, consider $\beta^-(\{1,3\}) = \boldsymbol{\chi}_2, \beta^-(\{3,5\}) = \boldsymbol{\chi}_2$, so $2\boldsymbol{\chi}_2 = (0,2)^\top$ is a lower bound on vertices $(1,5)$.

Consider the upper bounds on vertices $(4,5)$: there is $\beta^+(\{4,5\}) = \boldsymbol{\chi}_2$ as an upper bound on $(4,5)$. But notice, it is not the only upper bound: consider $\beta^+(\{2,5\}) = d_1, \beta^-(\{2,4\}) = d_2$. Then for any embedding Π with respect to $\boldsymbol{d}$, $\Pi(5) - \Pi(4) = ((\Pi(5) - \Pi(2)) - (\Pi(4) - \Pi(2)))^\top \boldsymbol{d} < (\beta^+(\{2,5\}) - \beta^-(\{4,5\}))^\top \boldsymbol{d} = (\boldsymbol{\chi}_1 - \boldsymbol{\chi}_2)^\top \boldsymbol{d}$. Thus, $\boldsymbol{\chi}_1 - \boldsymbol{\chi}_2 = (1,-1)^\top$ is also an upper bound on vertices $(4,5)$.

Example 3.10 shows how bounds can be generated by concatenating edges or null-edges that share the same endpoints. Next we define how to concatenate edges so that it results in upper bounds or lower bounds on pairs of vertices.

Definition 3.11. Given an alternating sequence $W = \langle w_0, e_1, w_1, \ldots, w_p \rangle$ and $w_0 < w_p$. Suppose

$$e_i \text{ is } \begin{cases} \text{an edge} & \text{if } w_{i-1} < w_i, \\ \text{an edge or a null-edge} & \text{if } w_{i-1} > w_i, \end{cases} \tag{3.2.5}$$

then we call W a (w_0, w_p)-*upper-bound-walk*, and define the *upper bound induced by* W as

$$\beta^+(W) = \sum_{i \in [p]: w_{i-1} < w_i} \beta^+(e_i) - \sum_{i \in [p]: w_{i-1} > w_i} \beta^-(e_i).$$

Suppose

$$e_i \text{ is } \begin{cases} \text{an edge} & \text{if } w_{i-1} > w_i, \\ \text{an edge or a null-edge} & \text{if } w_{i-1} < w_i, \end{cases}$$

then we call W a (w_0, w_p)-*lower-bound-walk*, and define the *lower bound induced by* W as

$$\beta^-(W) = \sum_{i \in [p]: w_i > w_{i-1}} \beta^-(e_i) - \sum_{i \in [p]: w_i < w_{i-1}} \beta^+(e_i).$$

Lemma 3.12. *For an alternating sequence of vertices and edges or null-edges* $W = \langle w_0, e_1, w_1 \ldots, w_p\rangle$, *where* $e_i = \{w_{i-1}, w_i\}$ *that satisfies (3.2.5). Then, if* $w_0 < w_p$, *then* $S_p(W)$ *is an upper bound on* w_0, w_p*; if* $w_p < w_0$*, then* $-S_p(W)$ *is a lower bound on* (w_p, w_0).

To simplify the notation in the following proof, consider $W = \langle w_0, \ldots, w_p\rangle$, define $S_q(W)$ as

$$S_q(W) = \sum_{i\in[q]:w_{i-1}<w_i} \beta^+(e_i) - \sum_{i\in[q]:w_{i-1}>w_i} \beta^-(e_i),$$

defined on all $q \in [p]$.

Proof: We proceed an inductive proof on the length p such that given W of length p that satisfies (3.2.5): if $w_0 < w_p$, then $S_p(W)$ is an upper bound on (w_0, w_p); if $w_p < w_0$, then $-S_p(W)$ is a lower bound on (w_p, w_0). When $p = 1$, by Lemma 3.9

1. if $w_0 < w_1$, then $\beta^+(e_1)$ is an upper bound on (w_0, w_1);

2. if $w_1 < w_0$, then $\beta^-(e_1)$ is a lower bound on (w_1, w_0).

Rewrite it in terms of $S_q(W)$: given $W = \langle w_0, e_1, w_1\rangle$, if $w_0 < w_1$, then $S_1(W) = \beta^+(e_1)$ is an upper bound w_0, w_1; if $w_1 < w_0$, then $-S_1(W) = \beta^-(e_1)$ is a lower bound on (w_1, w_0). So the base case holds.

Suppose that the inductive hypothesis holds for q, where $q < p$, such that, for any alternating sequence $W' = \langle w_0, e_1, w_1, \ldots, w_q\rangle$ that satisfies (3.2.5), $S_q(W')$ is an upper bound on (w_0, w_q) if $w_0 < w_q$ and $-S_q(W')$ is a lower bound on (w_q, w_0) if $w_q < w_0$.

Let $W = \langle w_0, \ldots, w_p\rangle$ be such that satisfies (3.2.5). Suppose that $w_0 < w_{p-1}$, then $S_{p-1}(W)$ is an upper bound on (w_0, w_{p-1}) by the inductive hypothesis. Recall Lemma 3.7:

1. if $w_0 < w_{p-1} < w_p$, then $\beta^+(e_p)$ is an upper bound on (w_{p-1}, w_p). So

$$S_{p-1}(W) + \beta^+(e_p) = S_p(W)$$

is an upper bound on (w_0, w_p) by Lemma 3.7 Item 1;

2. if $w_0 < w_p < w_{p-1}$, then $\beta^-(e_p)$ is a lower bound on (w_p, w_{p-1}). So

$$S_{p-1}(W) - \beta^-(e_p) = S_p(W)$$

is an upper bound on (w_0, w_p) by Lemma 3.7 Item 3;

3. if $w_p < w_0 < w_{p-1}$, then $\beta^-(e_p)$ is a lower bound on (w_p, w_{p-1}). So

$$\beta^-(e_p) - S_{p-1}(W) = -S_p(W)$$

is a lower bound on (w_p, w_0) by Lemma 3.7 Item 3.

Next, suppose $w_{p-1} < w_0$, then $-S_{p-1}$ is a lower bound on (w_{p-1}, w_0) by hypothesis. Again consider Lemma 3.7:

1. if $w_{p-1} < w_0 < w_p$, then $\beta^+(e_p)$ is an upper bound on (w_{p-1}, w_p). So

$$\beta^+(e_p) - (-S_{p-1}(W)) = S_p(W)$$

is an upper bound on (w_0, w_p) by Lemma 3.7 Item 2;

2. if $w_{p-1} < w_p < w_0$, then $\beta^+(e_p)$ is an upper bound on (w_{p-1}, w_p). So

$$(-S_{p-1}(W)) - \beta^+(e_p) = -S_p(W)$$

is a lower bound on (w_p, w_0) by Lemma 3.7 Item 2;

3. if $w_p < w_{p-1} < w_0$, then $\beta^-(e_p)$ is a lower bound on (w_p, w_{p-1}). So

$$(-S_{p-1}(W)) + \beta^-(e_p) = -S_p(W)$$

is a lower bound on (w_p, w_0) by Lemma 3.7 Item 1.

The above 6 cases conclude all possible orders of w_0, w_{p-1}, and w_p: Either $w_0 < w_p$, then $S_p(W)$ is an upper bound on (w_0, w_p); or $w_p < w_0$, then $-S_p(W)$ is a lower bound on (w_p, w_0). This shows that the inductive hypothesis holds when the length is p. $\square$

Given an alternating sequence $W = \langle w_0, \ldots, w_p \rangle$ that satisfies (3.2.5) and suppose $w_p < w_0$, so $-S_p(W)$ is a lower bound on (w_p, w_0). Define the *reverse of* W, $W^{\leftarrow}$, as the following and relabel the sequence as:

$$\begin{aligned} W^{\leftarrow} &= \langle w_p, e_p, w_{p-1}, \ldots, w_1, e_1, w_0 \rangle \\ &= \langle x_0, f_1, x_1, \ldots, x_{p-1}, f_p, x_p \rangle, \end{aligned}$$

where $x_i = w_{p-i}$ and $\{x_{i-1}, x_i\} = f_i = e_{p-i+1} = \{w_{p-i+1}, w_{p-i}\}$, and thus

$$f_i = \begin{cases} \text{an edge} & \text{if } x_{i-1} > x_i, \\ \text{an edge or a null-edge} & \text{if } x_{i-1} < x_i. \end{cases}$$

Notice that we only relabelled W, so Lemma 3.12 still holds. Therefore, when $-S_p(W)$ is a lower bound on (w_p, w_0); or equivalently, $-S_p(W)$ a lower bound on (x_0, x_p):

$$\begin{aligned}-S_p(W) &= \sum_{i\in[p]:w_{i-1}>w_i} \beta^-(e_i) - \sum_{i\in[p]:w_{i-1}<w_i} \beta^+(e_i) \\ &= \sum_{j\in[p]:x_j>x_{j-1}} \beta^-(f_j) - \sum_{i\in[p]:x_j<x_{j-1}} \beta^+(f_j) = \beta^-(W^{\leftarrow}).\end{aligned}$$

Meanwhile, notice $W^{\leftarrow}$ satisfies the definition of a (x_0, x_p)-lower-bound-walk.

In the above, we argued how to concatenate edges and null-edges so that it gives another upper bound (or lower bound) on a pair of vertices (u, v), where $u < v$. Such concatenation is conventionally defined as a walk (in graph theory), but the alternating sequence is not exactly a walk due to the possible inclusion of null-edges. Thus, we distinguish the two kinds of sequences, namely, (w_0, w_p)-upper-bound-walk if it induces an upper bound on (w_0, w_p), or (w_0, w_p)-lower-bound-walk if it induces a lower bound on (w_0, w_p), as in Definition 3.11 and Lemma 3.12.

For abbreviation, we omit the (u, v) pair before upper- or lower-bound-walks if the context is clear.

By Lemma 3.12, when W is an (u, v)-upper-bound-walk, $\beta^+(W)$ is an upper bound on (u, v); when W is an (u, v)-lower-bound-walk, then $\beta^-(W)$ is a lower bound on (u, v). Here are some examples of upper- and lower-bound-walk in examples below, and an intuitive picture in Figure 3.4.

Example 3.13. Use the matrix in Example 1.3 again, presented in Figure 3.4. Observe that the alternating sequence $W_{4,5} = \langle 4, \{4, 2\}, 2, \{2, 5\}, 5\rangle$ is a $(4, 5)$-upper-bound-walk. And notice $\beta^+(W_{4,5}) = -\beta^-(\{4, 2\}) + \beta^+(\{2, 5\}) = -(0, 1)^\top + (1, 0)^\top = (1, -1)^\top$ is a lower bound on $(4, 5)$.

Observe that the alternating sequence $W_{4,1} = \langle 4, \{4, 5\}, 5, \{5, 1\}, 1\rangle$ satisfies (3.2.5), where $\{5, 1\}$ is a null-edge. Then $W^{\leftarrow}_{4,1} = \langle 1, \{5, 1\}, 5, \{4, 5\}, 4\rangle$ is a $(1, 4)$-lower-bound-walk and $\beta^-(W^{\leftarrow}_{4,1}) = -S_2(W_{4,1}) = -(-(1, 0)^\top + (0, 1)^\top) = (1, -1)^\top$ is a lower bound on $(1, 4)$.

Also, notice that an alternating sequence $W = \langle w_0, \dots, v\rangle$ may be both an upper- and a lower-bound-walk if all the e_i involved in W are edges (no null-edges).

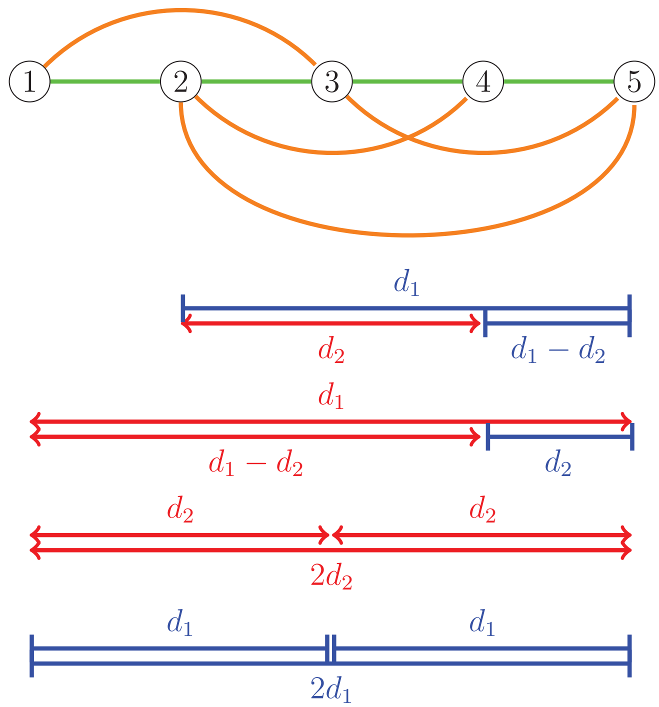

Figure 3.4: The graph of matrix A with induced bounds of bound-walks

Example 3.14. We repeat matrix $A = (a_{i,j})$ in Example 1.3 again:

$$(a_{i,j}) = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 2 & 2 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 1 & 2 & 2 \end{pmatrix} \end{array}$$

The alternating sequence $W = \langle 1, \{1,3\}, 3, \{3,5\}, 5\rangle$ is both a $(1,5)$-upper-bound-walk and a $(1,5)$-lower-bound-walk. So $\beta^+(W) = \beta^+(\{1,3\}) + \beta^+(\{3,5\}) = (1,0)^\top + (1,0)^\top = (2,0)^\top$, and $\beta^-(W) = \beta^-(\{1,3\}) + \beta^-(\{3,5\}) = (0,1)^\top + (0,1)^\top = (0,2)^\top$. These two examples are also presented in Figure 3.4.

Definition 3.15. Given two alternating sequences $W_1 = \langle u, \dots, v\rangle$ and $W_2 = \langle v, \dots, w\rangle$, denote $W_1 + W_2 = \langle u, \dots, v, \dots, w\rangle$ as the *concatenation of* W_1, W_2.

Remark 3.16. Let W be an alternating sequence and write $W = W_1 + W_2$, where $W = \langle u, \dots, s, \dots, v\rangle$ and $W_1 = \langle u, \dots, s\rangle, W_2 = \langle s, \dots, v\rangle$. Then W is a (u,v)-upper-bound-walk if and only if one of the following case holds for W_1, W_2, u, s, v:

1. W_1 is a (u, s)-upper-bound-walk, W_2 is an (s, v)-upper-bound-walk, and $u < s < v$;

2. W_1 is a (u, s)-upper-bound-walk, $W_2^{\leftarrow}$ is a (v, s)-lower-bound-walk, and $u < v < s$;

3. $W_1^{\leftarrow}$ is an (s, u)-lower-bound-walk, W_2 is an (s, v)-upper-bound-walk, and $s < u < v$.

Symmetrically with the same notation, W is a (u, v)-lower-bound-walk if and only if one of the following holds

4. W_1 is a (u, s)-lower-bound-walk, W_2 is an (s, v)-lower-bound-walk, and $u < s < v$;

5. W_1 is a (u, s)-lower-bound-walk, $W_2^{\leftarrow}$ is a (v, s)-upper-bound-walk, and $u < v < s$;

6. $W_1^{\leftarrow}$ is an (s, u)-upper-bound-walk, W_2 is an (s, v)-lower-bound-walk, and $s < u < v$.

Further, in each of the above 6 cases,

1. by Lemma 3.7 Item 1, $\beta^+(W) = \beta^+(W_1) + \beta^+(W_2)$ is an upper bound on (u, v);

2. by Lemma 3.7 Item 3, $\beta^+(W) = \beta^+(W_1) - \beta^-(W_2)$ is an upper bound on (u, v);

3. by Lemma 3.7 Item 2, $\beta^+(W) = \beta^+(W_2) - \beta^-(W_1)$ is an upper bound on (u, v);

4. by Lemma 3.7 Item 1, $\beta^-(W) = \beta^-(W_1) + \beta^-(W_2)$ is a lower bound on (u, v);

5. by Lemma 3.7 Item 3, $\beta^-(W) = \beta^-(W_1) - \beta^+(W_2)$ is a lower bound on (u, v);

6. by Lemma 3.7 Item 2, $\beta^-(W) = \beta^-(W_2) - \beta^+(W_1)$ is a lower bound on (u, v).

3.3 A Sufficient and Necessary Condition

Section 3.2 introduced the necessary concepts to state the main theorem of this thesis, that is: if a Robinson matrix has a uniform embedding with respect to threshold vector $\boldsymbol{d}$, then all the bounds that are implied by inequality system (3.1.1) "agree" with each other. It is natural to state an implication on the existence of a uniform embedding: if matrix has a uniform embedding, then there exists a threshold vector $\boldsymbol{d}$ so that all lower bounds $\boldsymbol{a}$ induced by (u,v)-lower-bound-walks "agree" with all upper bounds $\boldsymbol{b}$ induced by (u,v)-upper-bound-walks. Notice that the definition of bounds is independent from the threshold vector, but for any $\boldsymbol{d}$, the uniform embedding needs to satisfy that for any lower bound on (u,v), $\boldsymbol{a}$, and any upper bound on (u,v), $\boldsymbol{b}$, $\boldsymbol{a}^\top \boldsymbol{d} < \boldsymbol{b}^\top \boldsymbol{d}$.

As it turns out, this condition is also sufficient to show that the given matrix has a uniform embedding. Further, we only need to consider the *lower-* and *upper-bound-paths*, defined as lower- and upper-bound-walks containing no repeating vertices. We denote the set of (u,v)-lower-bound-paths as $\mathfrak{L}_{u,v}$ and the set of (u,v)-upper-bound-paths as $\mathfrak{U}_{u,v}$. Consider the inequality system:

Condition 3.17. Let $A \in \mathcal{S}^n[k]$, for all $u,v \in [n], u < v$, for all upper bounds $\boldsymbol{b} = \beta^+(W_1)$ where $W_1 \in \mathfrak{U}_{u,v}$ and for all lower bounds $\boldsymbol{a} = \beta^-(W_2)$ where $W_2 \in \mathfrak{L}_{u,v}$,

$$\boldsymbol{a}^\top \boldsymbol{d} < \boldsymbol{b}^\top \boldsymbol{d}. \tag{3.3.1}$$

Later, we abbreviate $\beta^+(\mathfrak{U}_{u,v})$ and $\beta^+(\mathfrak{L}_{u,v})$ for the set of all upper bounds and lower bounds on (u,v) induced by any upper- or lower-bound-paths. With Condition 3.17, we state our main result.

Theorem 3.18. *Given $A \in \mathcal{S}^n[k]$, A has an uniform embedding if and only if there exists $\boldsymbol{d} \in \mathbb{D}_k$ satisfying Condition 3.17.*

We can prove the necessity of Theorem 3.18 without other tools:

Proof of the forward implication of Theorem 3.18: Suppose that $A \in \mathcal{S}^n[k]$ has a uniform embedding. By Theorem 3.1, there exists a uniform embedding Π with respect to $\boldsymbol{d} \in \mathbb{D}_k$ that satisfies inequality system (3.1.1). Then, for all $u,v \in [n], u < v$, for any (u,v)-lower- and (u,v)-upper-bound-walks $W_1 \in \mathfrak{L}_{u,v}, W_2 \in \mathfrak{U}_{u,v}$, the induced lower and upper bounds $\boldsymbol{a} = \beta^-(W_1), \boldsymbol{b} = \beta^+(W_2)$ are implied lower and upper bounds on (u,v), *i.e.*, for all uniform embeddings Π with respect to $\boldsymbol{d}$, $\boldsymbol{a}^\top \boldsymbol{d} < \Pi(v) - \Pi(u) < \boldsymbol{b}^\top \boldsymbol{d}$. Therefore, omit the embedding Π, $\boldsymbol{a}^\top \boldsymbol{d} < \boldsymbol{b}^\top \boldsymbol{d}$ for any induced bounds of lower- and upper-bound-paths. $\square$

For the converse, we will obtain an iterative procedure to calculate the mapping Π that satisfies Condition 3.17 in Section 3.4. However, we need to first prove the Condition 3.17 stated in terms of "paths" implies that Condition 3.17 holds when stated in terms of "walks".

3.3.1 Cycles and Paths

In Section 3.2 we saw how walks can be used to generate new inequalities (*i.e.*, bounds) that are implied by the inequality system (3.1.1). In this section, we show that, for the existence of a uniform embedding, we need only to consider paths. A (u,v)-upper or (u,v)-lower-bound-walk $W = \langle u = w_0, w_1, \ldots, w_p = v\rangle$ is an *upper- or lower-bound-cycle* if $u = v$ and W contains no other repeated vertices.

Note that the order in which the cycle is traversed determines whether it is an upper- or lower-bound-cycle. Let C be an upper-bound-cycle, then notice $C^{\leftarrow}$ is a lower-bound-cycle: this gives $\beta^+(C) = -\beta^-(C^{\leftarrow})$.

Lemma 3.19. *Let $A \in \mathcal{S}^n[k]$ and $\boldsymbol{d} \in \mathbb{D}_k$. Let $C = \langle u_1, \ldots, u_p\rangle$, $u_1 = u_p$, be an upper-bound-cycle. If $\boldsymbol{d} \in \mathbb{D}_k$ satisfies Condition 3.17, then $\beta^+(C)^\top \boldsymbol{d} > 0$.*

Proof: Write $u = u_1 = u_p$. Let $v = u_i$ for some $1 < i < p$, and suppose $u < v$. Write $W = W_1 + W_2$, where $W_1 = \langle u = u_1, u_2, \ldots, u_i = v\rangle, W_2 = \langle v = u_i, u_{i+1} \ldots, u_{p-1}, u_p = u\rangle$. By Lemma 3.12, $\beta^+(W_1)$ is an upper bound on (u,v), and $\beta^-(W_2^{\leftarrow})$ is a lower bound on (u,v). By Remark 3.16,

$$\beta^+(C) = \beta^+(W_1) - \beta^-(W_2^{\leftarrow})$$

Then, by the choice of $\boldsymbol{d}$, $\beta^-(W_2^{\leftarrow})^\top \boldsymbol{d} < \beta^+(W_1)^\top \boldsymbol{d}$ and thus, $\beta^+(C)^\top \boldsymbol{d} > 0$.

The same logic applies when $v < u$. Write $W_1 = \langle v, \ldots, u_1\rangle$ and $W_2 = \langle v, \ldots, u_p\rangle$, so that $C = W_1^{\leftarrow} + W_2$ by Remark 3.16. Therefore, $\beta^+(C) = \beta^+(W_2) - \beta^-(W_1^{\leftarrow})$. By the choice of $\boldsymbol{d}$, $\beta^+(W_2)^\top \boldsymbol{d} > \beta^-(W_1)^\top \boldsymbol{d}$ and thus $\beta^+(C)^\top \boldsymbol{d} > 0$. □

Combining Lemma 3.19 and $\beta^+(C) = -\beta^-(C^{\leftarrow})$ gives $\beta^-(C^{\leftarrow})^\top \boldsymbol{d} < 0$ if C is a lower-bound-cycle.

Remark 3.20. Let $W = \langle w_0, \ldots, w_p\rangle$ be a (w_0, w_p)-upper-bound-walk with $w_0 < w_p$. Decompose $W = W_1 + C + W_2$ where $W_1 = \langle u = w_0, \ldots, w_i\rangle, C = \langle w_i, \ldots, w_{i+l}\rangle, W_2 = \langle w_{i+l}, \ldots, w_p\rangle$ where $w_i = w_{i+l}$. Notice, W is an upper-bound-walk if and only if C is an upper-bound-cycle by Definition 3.11. By Definition 3.11, $W_1 + W_2$ is a (w_0, w_p)-upper-bound-walk. Then, by Remark 3.16,

$$\beta^+(W_1 + W_2) = \begin{cases} \beta^+(W_1) + \beta^+(W_2) & \text{if } w_0 < w_i < w_p, \\ \beta^+(W_1) - \beta^-(W_2^{\leftarrow}) & \text{if } w_0 < w_p < w_i, \\ \beta^+(W_2) - \beta^-(W_1^{\leftarrow}) & \text{if } w_i < w_0 < w_p. \end{cases}$$

Then, by Lemma 3.12, $\beta^+(W) = \beta^+(W_1 + W_2) + \beta^+(C)$; this means we can detach the cycles that are contained in a walk, and the remaining edges form another upper-bound-walk. The same logic applies when W is a (w_0, w_p)-lower-bound-walk: If W is a lower-bound-walk such that $W = W_1 + C + W_2$. Decompose $W = W_1 + C + W_2$, then $\beta^-(W) = \beta^-(W_1 + W_2) + \beta^-(C)$.

Lemma 3.21. *Let $A \in \mathcal{S}^n[k]$. For any (u,v)-upper-bound-walk W, there exists a (u,v)-upper-bound-path W' so that, for any $\boldsymbol{d} \in \mathbb{D}_k$, $\beta^+(W')^\top \boldsymbol{d} \leqslant \beta^+(W)^\top \boldsymbol{d}$. Similarly, for any (u,v)-lower-bound-walk W, there exists a (u,v)-lower-bound-path W' so that, for any $\boldsymbol{d} \in \mathbb{D}_k$, $\beta^-(W)^\top \boldsymbol{d} \leqslant \beta^-(W')^\top \boldsymbol{d}$.*

Proof: It is trivial when W is already a (u,v)-upper-bound-path. Suppose $W = \langle u = w_0, e_1, w_1, \ldots, w_p = v \rangle$ contains a cycle. Decompose the walk W into

$$\begin{aligned} W_1 &= \langle u = w_0, e_1, \ldots, w_i \rangle, \\ C &= \langle w_i, e_{i+1}, \ldots, w_{i+l} \rangle, \\ W_2 &= \langle w_{i+l}, e_{i+l+1}, \ldots, w_p = v \rangle. \end{aligned}$$

such that $w_i = w_{i+l}$. Denote $W' = W_1 + W_2$. As discussed in Remark 3.20, C is an upper-bound-walk, and $\beta^+(W) = \beta^+(W') + \beta^+(C)$. By Lemma 3.19, for any $\boldsymbol{d} \in \mathbb{D}_k$,

$$\beta^+(W)^\top \boldsymbol{d} = \beta^+(W')^\top \boldsymbol{d} + \beta^+(C)^\top \boldsymbol{d} > \beta^+(W')^\top \boldsymbol{d}.$$

Iteratively remove any cycle in the walk W, say $W^{(i)}$ is the lower-bound-walk obtained by removing the ith cycle in W (*i.e.*, $W' = W^{(1)}$). Then obtain a sequence of walks and inequalities:

$$\beta^+(W^{(q)})^\top \boldsymbol{d} < \cdots < \beta^+(W^{(1)})^\top \boldsymbol{d} < \beta^+(W)^\top \boldsymbol{d}.$$

where $W^{(q)}$ contains no repeating vertices and is an upper-bound-path.

The same logic applies to lower-bound-paths: suppose $W = W_1 + C + W_2$ is a lower-bound-walk, and thus C is a lower-bound-cycle. Then, $\beta^-(W) = \beta^-(W_1 + W_2) + \beta^-(C)$, and $\beta^-(C)^\top \boldsymbol{d} < 0$, so $\beta^-(W)^\top \boldsymbol{d} < \beta^-(W_1 + W_2)^\top \boldsymbol{d}$. Iteratively remove any cycle in the walk to obtain a lower-bound-path. □

Lemma 3.21 shows that we only need to consider the upper- and lower-bound-paths in Condition 3.17. In other words, fix a pair $u, v \in [n], u < v$, for any (u,v)-upper-bound-walk, we can obtain a (u,v)-upper-bound-path by removing any cycle it contains; similarly for (u,v)-lower-bound-walks. Let W_1 be a (u,v)-lower-bound-walk and W_2 be a (u,v)-upper-bound-walk, we obtain a (u,v)-lower-bound-path W_1' and a (u,v)-upper-bound-path W_2' from W_1 and W_2 by removing the cycles they contain. Then, for any $\boldsymbol{d} \in \mathbb{D}_k$ satisfies Condition 3.17,

$$\beta^-(W_1)^\top \boldsymbol{d} \leqslant \beta^-(W_1')^\top \boldsymbol{d} < \beta^-(W_2')^\top \boldsymbol{d} \leqslant \beta^+(W_2)^\top \boldsymbol{d}.$$

3.4 Finding a Uniform Embedding

In the last section, we stated a necessary condition on the threshold vector $\boldsymbol{d}$ if a matrix has a uniform embedding. It turns out that it is sufficient for a matrix to have a uniform embedding if the inequality system in Condition 3.17 has a solution. In this section, we prove the converse of Theorem 3.18 by proposing a formula to compute a uniform embedding $\Pi : [n] \to \mathbb{R}$ of a given Robinson matrix A, given $\beta^+(\mathfrak{U}_{u,v}), \beta^-(\mathfrak{L}_{u,v})$ of $A \in \mathcal{S}^n[k]$ and $\boldsymbol{d} \in \mathbb{D}_k$ that satisfies Condition 3.17. The procedure is simple, we iteratively evaluate the restricted interval of where the next vertex can be embedded base on all the previously embedded vertices, while it does not violate Definition 1.2. By getting the interval, we define the embedding of the vertex to be on the middle of the restricted interval.

We first put down the formula of the desired embedding Π.

Definition 3.22. Let $A \in \mathcal{S}^n[k]$ and suppose there is $\boldsymbol{d} \in \mathbb{D}_k$ so that it satisfies Condition 3.17. Define mapping Π on $[n]$ as the following:

$$\begin{aligned} \Pi(1) &= 0, \\ \Pi(v) &= (ub_v + lb_v)/2 \quad \text{for } 1 < v \leqslant n, \end{aligned} \tag{3.4.1}$$

where ub_v, lb_v are defined based on $v-1$ as:

$$\begin{aligned} ub_v &= \min_{i \in [v-1]} \left\{ \Pi(i) + \min\{\boldsymbol{b}^\top \boldsymbol{d} : \boldsymbol{b} \in \beta^+(\mathfrak{U}_{i,v})\} \right\}, \\ lb_v &= \max_{i \in [v-1]} \left\{ \Pi(i) + \max\{\boldsymbol{a}^\top \boldsymbol{d} : \boldsymbol{a} \in \beta^-(\mathfrak{L}_{i,v})\} \right\}, \end{aligned} \tag{3.4.2}$$

Notice that $\mathfrak{U}_{u,v}$ and $\mathfrak{L}_{u,v}$ denote the sets of (u,v)-upper- and (u,v)-lower-bound-paths, they contain finitely many elements since there are no repeated vertices. And therefore, ub_v and lb_v both can be attained. The initial $\Pi(1) = 0$ is justified as in Lemma 3.3. Then iteratively, where ub_v can be seen as the "evaluated minimum upper bound" and lb_v as the "evaluated maximum lower bound". We will show that Π defined as such is an uniform embedding of A with respect to $\boldsymbol{d}$, which is the proof of the converse implication of Theorem 3.18.

To prove that Π is an uniform embedding, we first show that Π is strictly increasing.

Lemma 3.23. *The map Π as in Definition 3.22 is strictly increasing. Precisely, $0 < \Pi(v-1) \leqslant lb_v < ub_v$ for each $v \in [n]$, and thus $\Pi(v-1) < \Pi(v)$ for all $v \in [n-1]$.*

Proof: We will show, inductively, that Π is strictly increasing on $[v]$, for $v \in [n]$, with $\Pi(v-1) \leqslant lb_v < ub_v$. For the base case, $\Pi(1)$ itself is strictly increasing is trivial.

Inductively, suppose Π is strictly increasing defined on $[v-1]$. Let u, w be the vertices attaining ub_v and lb_v respectively. Let $\boldsymbol{b}_{\min} \in \beta^+(\mathfrak{U}_{u,v})$ be such that $\boldsymbol{b}_{\min}^\top \boldsymbol{d} = \min\{\boldsymbol{b}^\top \boldsymbol{d} : \boldsymbol{b} \in \beta^+(\mathfrak{U}_{u,v})\}$, and let $\boldsymbol{a}_{\max} \in \beta^-(\mathfrak{L}_{w,v})$ be such that $\boldsymbol{a}_{\max}^\top \boldsymbol{d} = \max\{\boldsymbol{a}^\top \boldsymbol{d} : \boldsymbol{a} \in \beta^-(\mathfrak{L}_{w,v})\}$, so that we have,

$$ub_v = \Pi(u) + \boldsymbol{b}_{\min}^\top \boldsymbol{d} = \min_{i \in [v-1]} \left\{ \Pi(i) + \min\{\boldsymbol{b}^\top \boldsymbol{d} : \boldsymbol{b} \in \beta^+(\mathfrak{U}_{i,v})\} \right\},$$
$$lb_v = \Pi(w) + \boldsymbol{a}_{\max}^\top \boldsymbol{d} = \max_{j \in [v-1]} \left\{ \Pi(j) + \max\{\boldsymbol{a}^\top \boldsymbol{d} : \boldsymbol{a} \in \beta^-(\mathfrak{L}_{j,v})\} \right\}.$$

By the choice of $\boldsymbol{d}$ that satisfies Condition 3.17, then $\boldsymbol{a}_{\max}^\top \boldsymbol{d} < \boldsymbol{b}_{\min}^\top \boldsymbol{d}$.

Let W_B be a (u, v)-upper-bound-walk such that $\boldsymbol{b}_{\min} = \beta^+(W_B)$ and let W_A be a (w, v)-lower-bound-walk such that $\boldsymbol{a}_{\max} = \beta^-(W_A)$.

1. Suppose $u = w$, then $\Pi(u) = \Pi(w)$. Then, $\boldsymbol{a}_{\max}$ is a lower bound on (u, v) and $\boldsymbol{b}_{\min}$ is an upper bound on (u, v). By the choice of $\boldsymbol{d}$ that satisfies Condition 3.17, $\boldsymbol{a}_{\max}^\top \boldsymbol{d} < \boldsymbol{b}_{\min}^\top \boldsymbol{d}$, and thus $lb_v < ub_v$.

2. Suppose $u < w$. Then, by Remark 3.16, $W_B + W_A^{\leftarrow}$ is a (u, w)-upper-bound-walk. Notice, since $u < w$, $\Pi(w)$ is defined after $\Pi(u)$, then the evaluation of ub_w in Definition 3.22 involves $\Pi(u)$, namely, $ub_w \leqslant \Pi(u) + \min\{\boldsymbol{b}^\top \boldsymbol{d} : \boldsymbol{b} \in \beta^+(\mathfrak{U}_{u,w})\}$. Meanwhile, by the inductive hypothesis, $lb_w < ub_w$, and thus $\Pi(w) = (ub_w + lb_w)/2 < ub_w$. Combine the above two inequalities and notice $W_B + W_A^{\leftarrow} \in \mathfrak{U}_{u,v}$, then
$$\Pi(w) < ub_w \leqslant \Pi(u) + \beta^+(W_B + W_A^{\leftarrow})^\top \boldsymbol{d},$$
which gives $\Pi(w) - \Pi(u) < \beta^+(W_B + W_A^{\leftarrow})^\top \boldsymbol{d}$. Then by Remark 3.16, $\beta^+(W_B + W_A^{\leftarrow}) = \beta^+(W_B) - \beta^-(W_A^{\leftarrow}) = \boldsymbol{b}_{\min} - \boldsymbol{a}_{\max}$, where the last inequality holds by the definition of ub_w. So
$$lb_v = \Pi(w) + \boldsymbol{a}_{\max}^\top \boldsymbol{d} < \Pi(u) + \boldsymbol{b}_{\min}^\top \boldsymbol{d} = ub_v.$$

3. Suppose $w < u$. Then by Remark 3.16, $W_A + W_B^{\leftarrow}$ is an (w, u)-lower-bound-walk. In the case of $w < u$, $\Pi(u)$ is defined after $\Pi(w)$, then $lb_u \geqslant \Pi(w) + \max\{\boldsymbol{a}^\top \boldsymbol{d} : \boldsymbol{a} \in \beta^-(\mathfrak{L}_{u,w})\}$. By the inductive hypothesis and Definition 3.22, $lb_u < ub_u$, and thus $lb_u < \Pi(u) = (ub_u + lb_u)/2$. Combine the inequalities and notice that $W_A + W_B^{\leftarrow} \in \mathfrak{L}_{w,u}$,
$$\Pi(w) + \beta^-(W_A + W_B^{\leftarrow})^\top \boldsymbol{d} \leqslant lb_u < \Pi(u)$$

Remark 3.16 gives $\beta^-(W_A + W_B^{\leftarrow}) = \boldsymbol{a}_{\max} - \boldsymbol{b}_{\min}$, then

$$lb_v = \Pi(w) + \boldsymbol{a}_{\max}^\top \boldsymbol{d} < \Pi(u) + \boldsymbol{b}_{\min}^\top \boldsymbol{d} = ub_v.$$

The above three cases conclude that $lb_v < ub_v$. Recall the fact that $\mathbf{0}$ is a lower bound on any pair u, v: by the choice of $\boldsymbol{d}$, for all $0 = \mathbf{0}^\top \boldsymbol{d} \leqslant \boldsymbol{a}^\top \boldsymbol{d}$ for any $\boldsymbol{a} \in \beta^-(\mathfrak{L}_{v-1,v})$. By the inductive hypothesis, we have that Π is strictly increasing on $[v-1]$, so it suffices to show that $\Pi(v-1) < \Pi(v)$ to conclude that Π is strictly increasing on $[v]$.

$$\begin{aligned}\Pi(v-1) &= \Pi(v-1) + \mathbf{0}^\top \boldsymbol{d} \\ &\leqslant \Pi(v-1) + \min\{\boldsymbol{a}^\top \boldsymbol{d} : \boldsymbol{a} \in \beta^-(\mathfrak{L}_{v,v-1})\} = lb_v.\end{aligned} \tag{3.4.3}$$

Therefore, combining all inequalities, we have

$$\Pi(v-1) \leqslant lb_v < (lb_v + ub_v)/2 = \Pi(v) < ub_v,$$

which means the inductive hypothesis holds for case v. □

Lemma 3.24. *[The converse implication of Theorem 3.18] Given Robinson matrix $A \in \mathcal{S}^n[k]$, and let Π be defined as in Definition 3.22. Then Π is a uniform embedding of A that satisfies the inequality system (3.1.1).*

Namely, given $A = (a_{i,j}) \in \mathcal{S}^n[k]$ and let Π to be the embedding computed by Definition 3.22. We need to show that given $u < v$, it satisfies Definition 1.2,

$$a_{u,v} = t \iff d_{t+1} < \Pi(v) - \Pi(u) < d_t.$$

Proof: Let $u, v \in [n]$ with $u < v$, and $a_{u,v} = t \neq 0$ (so $\{u, v\}$ is an edge). Then $\langle u, v\rangle$ is a (u, v)-upper-bound-path, so $\beta^+(\{u, v\}) \in \beta^+(\mathfrak{U}_{u,v})$. By Definition 3.11, $\beta^+(u, v)^\top \boldsymbol{d} = d_t$. By Definition 3.22,

$$\begin{aligned}\Pi(v) < ub_v &= \min_{i \in [v-1]} \left\{\Pi(i) + \min\{\boldsymbol{b}^\top \boldsymbol{d} : \boldsymbol{b} \in \beta^+(\mathfrak{U}_{i,v})\}\right\}, \\ &\leqslant \Pi(u) + \min\{\boldsymbol{b}^\top \boldsymbol{d} : \boldsymbol{b} \in \beta^+(\mathfrak{U}_{u,v})\} \\ &\leqslant \Pi(u) + \beta^+(u, v)^\top \boldsymbol{d} \\ &= \Pi(u) + d_t.\end{aligned}$$

Rewrite and we obtain $\Pi(v) - \Pi(u) < d_t$. If $t = 0$, then the inequality $\Pi(v) - \Pi(u) < d_0 = \infty$ is trivially satisfied.

If $t = k$, then the inequality $\Pi(v) - \Pi(u) > 0$ is satisfied since Π is strictly increasing. If $0 \leqslant t < k$, then $\langle u, v\rangle$ is a (u, v)-lower-bound-path. So $\beta^-(\{u, v\}) \in \beta^-(\mathfrak{L}_{u,v})$, and

$$\begin{aligned}
\Pi(v) > lb_v &= \max_{i \in [v-1]} \left\{\Pi(i) + \max\{\boldsymbol{b}^\top \boldsymbol{d} : \boldsymbol{b} \in \beta^-(\mathfrak{L}_{i,v})\}\right\}, \\
&\geqslant \Pi(u) + \max\{\boldsymbol{b}^\top \boldsymbol{d} : \boldsymbol{b} \in \beta^-(\mathfrak{L}_{u,v})\} \\
&\geqslant \Pi(u) + \beta^-(u, v)^\top \boldsymbol{d} \\
&= \Pi(u) + d_{t+1}.
\end{aligned}$$

Rewrite and obtain $\Pi(v) - \Pi(u) > d_{t+1}$. $\square$

Thus, we have established that Π defined by Definition 3.22 is a uniform embedding of the given Robinson matrix A with respect to $\boldsymbol{d}$.

Chapter 4

Testing the Conditions

In Chapter 3, we discussed a sufficient and necessary condition, Condition 3.17, so that a Robinson matrix, in $\mathcal{S}^n[k]$, has a uniform embedding. Notice that Condition 3.17 itself is a system of inequalities, based on some $\boldsymbol{d}$, such that $\boldsymbol{a}^\top \boldsymbol{d} < \boldsymbol{b}^\top \boldsymbol{d}$, where $\boldsymbol{b}, \boldsymbol{a}$ are obtained by enumerating the set of all (u, v)-upper- and lower-bound-paths. In this chapter, we discuss the entire procedure of finding a uniform embedding, including the enumeration of all the upper- and lower-bound-paths, find the threshold vector $\boldsymbol{d} \in \mathbb{D}_k$, and finally compute the embedding Π. We employ a variation of the Floyd-Warshall algorithm to generate all the upper- and lower-bound-paths. Then we rewrite the inequality system based on upper- and lower-bound-path in terms of upper-bound cycles in order to find a solution for $\boldsymbol{d}$. We will also discuss a partial order on the set of $\mathbb{Z}^k$, which reduces the complexity of the algorithm. We will discuss the complexity of the entire program. Finally, we discuss a combinatorial algorithm to find a uniform embedding when $k = 2$.

4.1 Bound Generation: A Variation of the Floyd-Warshall Algorithm

This section will discuss an algorithm that enumerates all the (u, v)-upper- and lower-bound-paths. In 1962, the Floyd-Warshall algorithm ([9]) was invented to solve the all-pairs shortest path problem of a graph (that contains no negative cycles). Consider a graph $G = (V, E)$ with a weight function $w : E \to \mathbb{R}$. Denote the vertex set of size n in arbitrary order as $V = \{1, \dots, n\}$. The shortest path algorithm finds the minimum distance between each pair of vertices i, j, $D(i, j)$. The Floyd-Warshall algorithm iterates through $s \in V$, and calculate the minimum distance of each pair by considering the path that only contains intermediate vertices (the vertices that are not the two ends in a path) in a restricted vertex set $\{1, \dots, s\}$ for $s = 1, \dots, n$; such a path, written as $W = \langle i, \dots, j \rangle$, is referred to as a *$s$-path* or a *$s$-$(i, j)$-path*. The recursive relation is defined as the following: Initially, set $D(i, j) = w(\{i, j\})$ if $\{i, j\} \in E$, or $D(i, j) = \infty$ otherwise. Iteratively, update s from 1 to n:

$$D(i, j) = \min\{D(i, j), D(i, s) + D(s, j)\} \tag{4.1.1}$$

Without further analysis, the Floyd-Warshall algorithm is a dynamic programming algorithm with time complexity $O(n^3)$. In addition, the algorithm detects negative cycles by checking the sign of each $D(i,i)$ for $i \in V$.

4.1.1 Bound-Generation Algorithm

As mentioned, the necessary condition of Theorem 3.18, Condition 3.17, includes an enumeration of all (u,v)-upper- and (u,v)-lower-bound-paths. Therefore, we employ the strategy similar to Floyd-Warshall algorithm to enumerate all upper-bound-paths and lower-bound-paths of a Robinson matrix $A \in \mathcal{S}^n[k]$. Notice that the two algorithms are not the same. For each pair of vertices, instead of the weight function w, our "weight" function consists of β^+ and β^-, which induces upper bounds or lower bounds implied by inequalities (3.1.1). The Floyd-Warshall algorithm determines the path with smallest weight $D(i,j)$; at any step, it can decide which path is the shortest path. The Floyd-Warshall algorithm determines the best choice since the minimum weight of the paths, $D(i,j)$, is a real number. In our case, we cannot know which element in $\beta^+(\mathfrak{U}_{u,v})$ or $\beta^-(\mathfrak{L}_{u,v})$ contributes to the minimum or maximum evaluation in advance. Precisely, real numbers are totally ordered, but bounds are not always comparable. Therefore, we need to store all the upper- and lower-bound-paths or their induced upper or lower bounds. Later in the thesis, we will prove there is a partial order on the bounds (in fact, on $\mathbb{Z}^k$), so that we may discard some bounds. But essentially, the recording of bounds and the enumeration will cost more than the simple comparison as in the Floyd-Warshall algorithm.

We implement our algorithm in Algorithm 1. Briefly, given a Robinson matrix $A = (a_{u,v}) \in \mathcal{S}^n[k]$, Algorithm 1 performs the following operations and records upper-bound-paths in $\texttt{UBW}(u,v)$ and lower-bound-paths in $\texttt{LBW}(u,v)$.

1. Initialize $\texttt{UBW}(u,v)$ with $\{\langle u,v\rangle\}$ if $a_{u,v} \neq 0$, or initialize to empty set if $a_{u,v} = 0$; initialize $\texttt{LBW}(u,v)$ with $\{\langle u,v\rangle\}$.

2. Notice the idea from Floyd-Warshall algorithm that iteratively constructs the paths with intermediate vertices restricted to $[s]$ at each iteration s. We extend this principle to generate all upper- and lower-bound-paths. In analogy, define a s-(u,v)-*upper-bound-path* to be a (u,v)-upper-bound-path with its intermediate vertices restricted to $[s]$; define a s-(u,v)-*lower-bound-path* to be a (u,v)-lower-bound-path with its intermediate vertices restricted to $[s]$. At each iteration s,

Algorithm 1: Bound-Generation

input : Robinson matrix $A \in \mathcal{S}^n[k]$

output: Lookup tables `UBW`, `LBW` defined on $i, j \in [n], i < j$: where

$\texttt{UBW}(i, j) = \mathfrak{U}_{i,j}$,

$\texttt{LBW}(i, j) = \mathfrak{L}_{i,j}$.

```
for i ∈ [n] do
    for j = i, ..., n do
        if a_{i,j} ≠ 0 then UBW(i, j) ← {⟨i, j⟩} ;
        LBW(i, j) ← {⟨i, j⟩};
for s = 1, ..., n do
    for i = 1, ..., n do
        for j = i, ..., n and i ≠ s ≠ j do
            if i < s < j then
                foreach W_1 ∈ UBW(i, s) and W_2 ∈ UBW(s, j) do
                    Add-Walk-To(UBW(i, j), W_1 + W_2);
                foreach W_1 ∈ LBW(i, s) and W_2 ∈ LBW(s, j) do
                    Add-Walk-To(LBW(i, j), W_1 + W_2);
            else if i < j < s then
                foreach W_1 ∈ UBW(i, s) and W_2 ∈ LBW(j, s) do
                    Add-Walk-To(UBW(i, j), W_1 + W_2^←);
                foreach W_1 ∈ LBW(i, s) and W_2 ∈ UBW(j, s) do
                    Add-Walk-To(LBW(i, j), W_1 + W_2^←);
            else if s < i < j then
                foreach W_1 ∈ LBW(s, i) and W_2 ∈ UBW(s, j) do
                    Add-Walk-To(UBW(i, j), W_1^← + W_2);
                foreach W_1 ∈ UBW(k, i) and W_2 ∈ LBW(k, j) do
                    Add-Walk-To(LBW(i, j), W_1^← + W_2);
return UBW, LBW;
```

Algorithm 2: Add-Walk-To

input : A set of upper(lower)-bound-paths S and an upper-(lower)-bound-path W

if W *contains no repeating vertices* **then** $S' \leftarrow S' \cup \{W\}$;

for each pair $u, v \in [n]$, $u < v$, compute all s-(u, v)-upper- and s-(u, v)-lower-bound-paths, for $s = 1, \ldots, n$. In each iteration s, the concatenation of upper- and lower-bound-paths follows Remark 3.16.

4.1.2 The Correctness of Bound-Generation Algorithm

We denote the set of all s-(u, v)-upper-bound-paths as $\mathfrak{U}^s_{i,j}$ and the set of all s-(u, v)-lower-bound-paths as $\mathfrak{L}^s_{i,j}$. We prove the `Bound-Generation()` algorithm with a similar proof as the proof of the Floyd-Warshall algorithm.

Theorem 4.1 (Correctness of Algorithm 1)**.** *Given a Robinson matrix $A \in \mathcal{S}^n[k]$. Let* $\texttt{UBW}, \texttt{LBW}$ *be the returned tables of* $\texttt{Bound-Generation}(A)$*. Then* $\texttt{LBW}(i, j) = \mathfrak{L}_{i,j}$ *and* $\texttt{UBW}(i, j) = \mathfrak{U}_{i,j}$*.*

Proof: We proceed a proof by induction on the restricted intermediate vertex set $[s]$ for $0 \leqslant s \leqslant n$, such that $\texttt{UBW}(i, j) = \mathfrak{U}^s_{i,j}$ and $\texttt{LBW}(i, j) = \mathfrak{L}^s_{i,j}$ at each iteration s.

Recall that we defined $[0] = \emptyset$. Line 1-4 initializes $\texttt{UBW}(i, j) = \{\langle i, j\rangle\}$ if $a_{i,j} \neq 0$ or empty otherwise, and $\texttt{LBW}(i, j) = \{\langle i, j\rangle\}$. So $\texttt{UBW}(i, j) = \mathfrak{U}^0_{i,j}$ and $\texttt{LBW}(i, j) = \mathfrak{L}^0_{i,j}$ for all $i, j \in [n]$ for $i < j$, *i.e.*, they contain the upper- and lower-bound-paths that contain $[0]$, no intermediate vertices.

Inductively, suppose $\texttt{UBW}(i, j) = \mathfrak{U}^{s-1}_{i,j}$ and $\texttt{LBW}(i, j) = \mathfrak{L}^{s-1}_{i,j}$ at the end of iteration $s-1$. We will show that $\texttt{UBW}(i, j) = \mathfrak{U}^s_{i,j}$ and $\texttt{LBW}(i, j) = \mathfrak{L}^s_{i,j}$ at the end of iteration s.

1. We first show $\texttt{UBW}(i, j) \subseteq \mathfrak{U}^s_{i,j}$. Let $W \in \texttt{UBW}(i, j)$ at the end of iteration s. Notice that `Bound-Generation` does not remove elements from $\texttt{UBW}(i, j)$. We first suppose that $W \in \texttt{UBW}(i, j)$ at iteration $s - 1$ already, that is, W does not contain vertex s, then $W \in \mathfrak{U}^{s-1}_{u,v}$ by the inductive hypothesis. Note that $\mathfrak{U}^{s-1}_{i,j} \subseteq \mathfrak{U}^s_{i,j}$, thus $W \in \mathfrak{U}^s_{u,v}$.

   Next, suppose that W contains vertex s, which implies that W is added to $\texttt{UBW}(i, j)$ during iteration s. We write $W = \langle i, \ldots, s, \ldots, j\rangle$ by concatenation, $W = W_1 + W_2$ where $W_1 = \langle i, \ldots, s\rangle$ and $W_2 = \langle s, \ldots, j\rangle$. Then, depending on the order between i, j, s, either one of the following three cases holds:

(a) Line 10 generates W, $W_1 \in$ `UBW`(i,s) and $W_2 \in$ `UBW`(s,j), and $i < s < j$;
(b) Line 15 generates W, $W_1 \in$ `UBW`(i,s) and $W_2^{\leftarrow} \in$ `LBW`(j,s), and $i < j < s$;
(c) Line 20 generates W, $W_1^{\leftarrow} \in$ `LBW`(s,i) and $W_2 \in$ `UBW`(s,j), and $s < i < j$.

By the inductive hypothesis, `UBW`$(i,s) = \mathfrak{U}_{i,s}^{s-1}$, `LBW`$(s,i) = \mathfrak{L}_{s,i}^{s-1}$, `UBW`$(s,j) = \mathfrak{U}_{i,s}^{s-1}$, `LBW`$(j,s) = \mathfrak{L}_{j,s}^{s-1}$ at iteration $s-1$. This implies that W_1, W_2 or $W_1^{\leftarrow}, W_2^{\leftarrow}$, which appears in the above three cases, are $(s-1)$-upper- or $(s-1)$-lower-bound-paths. Then, by Remark 3.16, W is a (i,j)-upper-bound-walk. Note that W contains no repeating vertices since $W \in$ `UBW`(i,j), *i.e.*, `Add-Walk-To` checks that W contains no repeating vertices. And since W_1 and W_2 contain vertex s as the largest vertex , W is a s-(i,j)-upper-bound-path, *i.e.*, $W \in \mathfrak{U}_{i,j}^{s}$.

2. We then need to show $\mathfrak{U}_{i,j} \subseteq$ `UBW`(i,j). Let $W \in \mathfrak{U}_{u,v}^{s}$ be an s-(i,j)-upper-bound-path. If W does not contain vertex s, then $W \in \mathfrak{U}_{i,j}^{s-1}$. By the inductive hypothesis, $\mathfrak{U}_{i,j}^{s-1} =$ `UBW`(i,j) at iteration $s-1$, therefore, $W \in$ `UBW`(i,j) at iteration s since there are no deleted elements.

   Suppose W contains vertex s, then W can be decomposed as $W = W_1 + W_2$ where $W_1 = \langle i, \ldots, s\rangle$ and $W_2 = \langle s, \ldots, j\rangle$. Notice W_1 and W_2 do not contain s as an intermediate vertex since they are paths, and W_1 and W_2 contain no common vertices.

   Suppose $i < s < j$, then $W_1 \in \mathfrak{U}_{i,s}^{s-1}$ and $W_2 \in \mathfrak{U}_{s,j}^{s-1}$ by Remark 3.16. By the inductive hypothesis, `UBW`$(i,s) = \mathfrak{U}_{i,s}^{s-1}$ and `UBW`$(s,j) = \mathfrak{L}_{s,j}^{s-1}$ at iteration $s-1$. Then, at Line 10, W_1 and W_2 are enumerated, and since W_1 and W_2 are upper-bound-paths that contain no common vertices, `Add-Walk-To`(`UBW`$(i,j), W_1+W_2)$ appends $W_1 + W_2$ to `UBW`(i,j). Similarly, when $i < j < s$, $W_1 \in \mathfrak{U}_{i,s}^{s-1}, W_2^{\leftarrow} \in \mathfrak{L}_{j,s}^{s-1}$. By the inductive hypothesis, $\mathfrak{U}_{i,s} =$ `UBW`$(i,s), \mathfrak{L}_{j,s} =$ `LBW`(j,s) at iteration $s-1$, so W_1 and $W_2^{\leftarrow}$ will be enumerated and $W_1 + W_2^{\leftarrow}$ will be added to `UBW`(i,j) at Line 15. Finally, when $s < i < j$, $W_1^{\leftarrow} \in \mathfrak{L}_{s,i}^{s-1}$ and $W_2 \in \mathfrak{U}_{j,s}^{s-1}$. Then $\mathfrak{L}_{s,i} =$ `LBW`$(s,i), \mathfrak{U}_{s,j} =$ `UBW`(s,j) at iteration $s-1$ by the inductive hypothesis. Thus, $W_1^{\leftarrow} \in$ `LBW`$(s,i), W_2 \in$ `UBW`(s,j) will be enumerated and $W_1^{\leftarrow} + W_2$ will be added to `UBW`(i,j) at Line 20.

The similar logic applies to show that `LBW`$(i,j) = \mathfrak{L}_{i,j}^{s}$ at iteration s. Both directions conclude that $\mathfrak{U}_{i,j}^{s} =$ `UBW`(i,j) and $\mathfrak{L}_{i,j}^{s} =$ `LBW`(i,j) at the end of iteration s, and the inductive hypothesis holds for case s. At iteration $s = n$, when the program finishes, we have that `UBW`$(i,j) = \mathfrak{U}_{i,j}^{n} = \mathfrak{U}_{i,j}$ and `LBW`$(i,j) = \mathfrak{L}_{i,j}^{n} = \mathfrak{L}_{i,j}$, which was what we want. $\square$

4.2 A Partial Order on Bounds

In this section, we introduce a partial order on $\mathbb{Z}^k$, which is special on the bounds. Notice that in `Bound-Generation`, the algorithm generates all (u, v)-upper- and (u, v)-lower-bound-paths, $\mathfrak{U}_{u,v}, \mathfrak{L}_{u,v}$. However, in Condition 3.17, we find a solution for $\boldsymbol{d}$ with only the set of upper bounds induced by upper-bound-cycles; also in Definition 3.22, we calculate the uniform embedding using only the set of bounds that are induced by the bound-paths, $\beta^+(\mathfrak{U}_{u,v}), \beta^-(\mathfrak{L}_{u,v})$. This means that the upper bounds and lower bounds are all we need. We also notice that, if two (u, v)-upper-bound-walks, W_1 and W_2, contribute the same upper bound on (u, v), *i.e.*, $\beta^+(W_1) = \beta^+(W_2)$, we need not to record both W_1 and W_2 in $\mathtt{UBW}(u, v)$. The following example shows there are many bounds that can be ignored, in either Definition 3.22 or Condition 3.17.

Example 4.2. In Example 1.3 matrix A, bound $(0, 1)^\top$ is an upper bound on $(2, 3)$, and bound $(1, 0)^\top$ is an upper bound on $(3, 5)$ and $(2, 5)$. Then, the following upper bounds are implied by the system (3.1.1).

$$\left.\begin{array}{l}\Pi(3) - \Pi(2) < d_2 \\ \Pi(5) - \Pi(2) < d_1 \\ \Pi(5) - \Pi(3) < d_1\end{array}\right\} \Rightarrow \begin{array}{rl}(\Pi(5) - \Pi(3)) + (\Pi(3) - \Pi(2)) & \\ = \Pi(5) - \Pi(2) & < d_1 + d_2 \\ \text{and } \Pi(5) - \Pi(2) & < d_1\end{array}$$

So $(1, 1)^\top$ and $(1, 0)^\top$ are both bounds on $(2, 5)$. Observe that, since $d_1 > d_2 > 0$, $(1, 0)\boldsymbol{d} < (1, 1)\boldsymbol{d}$ for all $\boldsymbol{d} \in \mathbb{D}_2$. Apply these two bounds to Condition 3.17, if $\boldsymbol{d}$ satisfies that, for all lower bounds $\boldsymbol{a}$ on (u, v), we have that $\boldsymbol{a}^\top\boldsymbol{d} < (1, 0)\boldsymbol{d}$, then $\boldsymbol{a}^\top\boldsymbol{d} < (1, 1)\boldsymbol{d}$ is trivially satisfied.

We introduce the following definition of a partial order on $\mathbb{Z}^k$ that generalizes the above observation. We will use this definition to adjust the function `Bound-Generation` and `Add-Walk-To` to identify and remove all such redundant bounds.

Definition 4.3. Define relation $\preccurlyeq$ on $\mathbb{Z}^k$ as follows. Given any $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i) \in \mathbb{Z}^k$, $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$ if

$$\sum_{i=1}^{t} a_i \leqslant \sum_{i=1}^{t} b_i \qquad \text{for all } t \in [k].$$

Suppose $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$, we say $\boldsymbol{a}$ is *tighter* than $\boldsymbol{b}$, or $\boldsymbol{b}$ is *wider* than $\boldsymbol{a}$. Otherwise, $\boldsymbol{a}, \boldsymbol{b}$ are *incomparable*.

We show that this relation is actually a partial order on $\mathbb{Z}^k$.

Lemma 4.4. *Relation $\preccurlyeq$ defines a partial order on $\mathbb{Z}^k$.*

Proof: Let $\boldsymbol{a} = (a_i) \in \mathbb{Z}^k$, indeed $\sum_{i=1}^t a_i \leqslant \sum_{i=1}^t a_i$ for all $t \in [k]$. So $\preccurlyeq$ is reflexive.

Let $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i) \in \mathbb{Z}^k$. Suppose that $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$ and $\boldsymbol{b} \preccurlyeq \boldsymbol{a}$, then

$$\sum_{i=1}^t a_i \leqslant \sum_{i=1}^t b_i, \quad \sum_{i=1}^t b_i \leqslant \sum_{i=1}^t a_i \qquad \text{for all } t \in [k].$$

This gives $\sum_{i=1}^t a_i = \sum_{i=1}^t b_i$ for all t. Thus, $a_1 = b_1$ and for all t such that $1 < t \leqslant k$,

$$a_t = \sum_{i=1}^t a_i - \sum_{i=1}^{t-1} a_i = \sum_{i=1}^t b_i - \sum_{i=1}^{t-1} b_i = b_t.$$

So $\preccurlyeq$ is antisymmetric.

Let $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i), \boldsymbol{c} = (c_i) \in \mathbb{Z}^k$. Suppose $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$ and $\boldsymbol{b} \preccurlyeq \boldsymbol{c}$, then

$$\sum_{i=1}^t a_i \leqslant \sum_{i=1}^t b_i \leqslant \sum_{i=1}^t c_i \qquad \text{for all } t \in [k].$$

This follows the definition such that $\boldsymbol{a} \preccurlyeq \boldsymbol{c}$. So $\preccurlyeq$ is transitive. $\square$

In the following theorem, we will show that for arbitrary upper bounds $\boldsymbol{a}$ and $\boldsymbol{b}$, if $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$, then $\boldsymbol{b}$ is a "weaker" upper bound than $\boldsymbol{a}$, and this theorem identifies the "useful" upper and lower bounds in Definition 3.22 and Condition 3.17. As in Example 4.2, $(1, 0)^\top$ and $(1, 1)^\top$ are upper bounds on $(2, 5)$ both belong to $\beta^+(\mathfrak{U}_{2,5})$. In Definition 3.22 that computes the uniform embedding, we get ub_5 as the upper bound on the placement of $\Pi(5)$, and $ub_5 \leqslant \Pi(2) + (1, 0)^\top \boldsymbol{d} < \Pi(2) + (1, 1)^\top \boldsymbol{d}$. Therefore, $(1, 0)^\top$ might be used to determine $\Pi(5)$, but $(1, 1)^\top$ will never be used. Also consider in Condition 3.17, for any $\boldsymbol{a} = (a_1, a_2)^\top \in \beta^-(\mathfrak{L}_{2,5})$, we may make up upper-bound-cycles C_1, C_2 such that $\beta^+(C_1) = (1, 0)^\top - \boldsymbol{a} = (1 - a_1, -a_2)^\top$, and $\beta^+(C_2) = (1-a_1, 1-a_2)^\top$. If $\boldsymbol{d} = (d_1, d_2)^\top$ satisfies that $\beta^+(C_1)^\top \boldsymbol{d} = d_1 - a_1 d_1 - a_2 d_2 > 0$, then $\beta^+(C_2)^\top \boldsymbol{d} = d_1 - a_1 d_1 + d_2 - a_2 d_2 > 0$ since $d_2 > 0$. Therefore, if bounds $\boldsymbol{a}, \boldsymbol{b}$ have that $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$ for any $\boldsymbol{d} \in \mathbb{D}_k$, then $\boldsymbol{b}$ does not provide new information to either Definition 3.22 or Condition 3.17. The following theorem provides a simple way to determine this relation. Then, after proving this theorem, we will modify the `Bound-Generation` algorithm with this partial order to keep only the bound-paths that induce minimal bounds in `UBW` and maximal bounds in `LBW`. Also notice, this theorem provides a second proof that $\preccurlyeq$ is a partial order.

Theorem 4.5. *Let* $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}^k$, *then* $\boldsymbol{a} \preccurlyeq \boldsymbol{b} \iff \boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$ *for all* $\boldsymbol{d} \in \mathbb{D}_k$.

We need some supplementary definitions and lemmas to prove this theorem. We will devote the rest of this section to prove Theorem 4.5. The intuition of proving this theorem is to construct a "buffer" bound $\boldsymbol{c} \in \mathbb{Z}^k$ so that $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{c}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$ holds for any $\boldsymbol{d} \in \mathbb{D}_k$. The construction is very technical; therefore, I shall show one example.

Example 4.6. Consider $\heartsuit, \diamondsuit, \clubsuit, \spadesuit$ are four objects with weights d_1, d_2, d_3, d_4 where $d_1 > d_2 > d_3 > d_4 > 0$. Then, a collection of a_1 number of $\heartsuit$, a_2 number of $\diamondsuit$, a_3 number of $\clubsuit$, and a_4 number of $\spadesuit$ together has weight $\boldsymbol{a}^\top \boldsymbol{d}$, where $\boldsymbol{a} = (a_1, a_2, a_3, a_4)^\top$ and $\boldsymbol{d} = (d_1, d_2, d_3, d_4)^\top$. We consider two collections of objects, and arrange them with into four slots:

Collection 1	$\heartsuit\heartsuit\heartsuit\heartsuit$	$\diamondsuit\diamondsuit$	$\clubsuit\clubsuit\clubsuit$	$\spadesuit\spadesuit$
Weights	$4d_1$	$2d_2$	$3d_3$	$2d_4$
Collection 2	$\heartsuit\heartsuit\heartsuit$	$\diamondsuit$	$\clubsuit$	$\spadesuit\spadesuit\spadesuit\spadesuit\spadesuit$
Weights	$3d_1$	d_2	d_3	$5d_4$

Then, we rearrange the collection 2 as the following

Collection 1	$\heartsuit\heartsuit\heartsuit\heartsuit$	$\diamondsuit\diamondsuit$	$\clubsuit\clubsuit\clubsuit$	$\spadesuit\spadesuit$
Weights	$4d_1$	$2d_2$	$3d_3$	$2d_4$
Collection 2 rearranged	$\heartsuit\heartsuit\heartsuit\spadesuit$	$\diamondsuit\spadesuit$	$\clubsuit\spadesuit$	$\spadesuit\spadesuit$
Weights	$3d_1 + d_4$	$d_2 + d_4$	$d_3 + d_4$	$2d_4$

Then, notice that the weight in each slot in collection 1 is greater than collection 2, since $d_i > d_4$ for any $i < 3$. We construct another collection 3 from collection 2 so that, in each slot, replace $\spadesuit$ by another type:

Collection 1	$\heartsuit\heartsuit\heartsuit\heartsuit$	$\diamondsuit\diamondsuit$	$\clubsuit\clubsuit\clubsuit$	$\spadesuit\spadesuit$
Weights	$4d_1$	$2d_2$	$3d_3$	$2d_4$
Collection 3	$\heartsuit\heartsuit\heartsuit\heartsuit$	$\diamondsuit\diamondsuit$	$\clubsuit\clubsuit$	$\spadesuit\spadesuit$
Weights	$4d_1$	$2d_2$	$2d_3$	$2d_4$
Collection 2 rearranged	$\heartsuit\heartsuit\heartsuit\spadesuit$	$\diamondsuit\spadesuit$	$\clubsuit\spadesuit$	$\spadesuit\spadesuit$
Weights	$3d_1 + d_4$	$d_2 + d_4$	$d_3 + d_4$	$2d_4$

Then, notice that the weight of each slot in collection 1 is higher (heavier) than or equal to the corresponding slot in collection 3; therefore the total weight of collection 1 is heavier than collection 3. Also, notice that we constructed collection 3 from collection 2 by replacing $\spadesuit$ by something heavier, *i.e.*, $\heartsuit$, $\diamondsuit$, or $\clubsuit$; therefore, the total weight of collection 3 is heavier than collection 2. Let $\boldsymbol{b} = (b_i), \boldsymbol{a} = (a_i), \boldsymbol{c} = (c_i) \in \mathbb{Z}^4$ denotes the number of $\heartsuit, \diamondsuit, \clubsuit, \spadesuit$ in each collection 1, 2, 3; then weight comparison is $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{c}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$.

The above example shows an intuition of the proof of Theorem 4.5. Suppose two vectors $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i) \in \mathbb{Z}^k$ have that $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$, then $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$ is obvious if $a_i \leqslant b_i$ for all $i \in [k]$. If the two vectors cannot be compared component-wise (*i.e.*, $a_i \leqslant b_i$ for all $i \in [k]$), then we rearrange the components and construct a "buffer" vector $\boldsymbol{c} = (c_i)$ (such as collection 3), so that

- $c_i \leqslant b_i$ for all $i \leqslant k$ and
- we may obtain $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{c}^\top \boldsymbol{d}$ easily, according to the construction.

Further, we will generalize the idea of the collection 3 in Example 4.6 so that, iteratively for every sub-vector (*i.e.*, the vector formed by the first t components), we will construct a "buffer" vector that satisfies the above two conditions. In the following content, we will denote function $[x]_+$ to be

$$[x]_+ = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{if } x \leqslant 0. \end{cases}$$

Definition 4.7. Let $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i) \in \mathbb{Z}^k$. Suppose $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$, define $\boldsymbol{c}$ with the following steps.

Iteratively, for $t = 1, \ldots, k$, define $\{c_{t,i}\}_{i \in [t]}$ as follows. Define $e_t = [a_t - b_t]_+$. Define $c_{t,t} = a_t - e_t$. Then, define two sequences, $\{e_{t,i}\}_{i \in [t-1]}$ and $\{f_{t,i}\}_{i \in [t]}$, as the following. Define $f_{t,1} = e_t$. Then, for $i = 1, \ldots, t-1$, define $e_{t,i}$ with $f_{t,i}$ and define $f_{t,i+1}$ with $e_{t,i}$:

$$\begin{aligned} e_{t,i} &= \min\{f_{t,i}, b_i - c_{t-1,i}\}, \\ f_{t,i+1} &= f_{t,i} - e_{t,i}. \end{aligned} \tag{4.2.1}$$

Define $c_{t,i} = c_{t-1,i} + e_{t,i}$ for all $1 \leqslant i < t$. Define $\boldsymbol{c} = (c_{k,i})_{i \in [k]}$.

Using Example 4.6 as an example. Consider $\boldsymbol{a}$ corresponds to collection 2 and $\boldsymbol{b}$ corresponds to collection 1. Notice that, for $t = 1, 2, 3$, the procedure will construct $c_{t,i} = a_i$ for $i < t$. When $t = 4$, we have that $5 = a_4 > b_4 = 2$. Then, define $c_{4,4} = 2$ and $e_4 = 3$, *i.e.*, e_4 represents how many item we need to re-distribute to the previous slots. Then, $e_{4,i}$ represents the number of items that is assigned to each slot i, and we re-distribute them iteratively from 1 to $t-1$: we denote $f_{t,i}$ as "how many items remain that need to be re-distributed before slot i"; naturally, $f_{t,1} = e_t$ is the total number of items that needs to be re-distributed. In Example 4.6, $e_{4,1} = e_{4,2} = e_{4,3} = 1$. After the re-distribution, we define a new collection as collection 3: $c_{4,i} = c_{3,i} + e_{4,i}$. Finally, the redistribution follows the rule such that each $c_{4,i} \leqslant b_i$ so that the comparison between collection 1 and 3 remains simple.

Lemma 4.8. *Let $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i) \in \mathbb{Z}^k$ and suppose $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$. Following the notations in Definition 4.7, for all $t \in [k]$, the following holds:*

1. *Sequence $\{f_{t,i}\}_{i\in[t]}$ is a non-negative and decreasing sequence.*
2. *For all $i \in [t]$, $c_{t,i} \leqslant b_i$.*

Proof: Let $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i) \in \mathbb{Z}^k$ and suppose $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$. Following the notations in Definition 4.7. We will give a proof by induction such that, for all $t \in [k]$, $\{f_{t,i}\}$ is a non-negative and decreasing sequence; and for all $i \in [t]$, we have that $c_i \leqslant b_i$. Consider the base case $t = 1$. The assumption that $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$ gives that $a_1 \leqslant b_1$. Therefore, $e_t = [a_1 - b_1]_+ = 0$ and $c_{1,1} = a_1 \leqslant b_1$. Since $t = 1$ and $e_t = f_{1,1} = 0$, the sequence with one element, $\{f_{1,1}\}$, is non-negative and decreasing is vacuously satisfied.

Inductively, for any $t > 1$, suppose that

Assumption 4.9.

- *Sequence $\{f_{t-1,i}\}_{i\in[t-1]}$ is a non-negative and decreasing sequence.*
- *For all $i \in [t-1]$, $c_{t-1,i} \leqslant b_i$.*

We will show that

- Sequence $\{f_{t,i}\}_{i\in[t]}$ is a non-negative and decreasing sequence.
- For all $i \in [t]$, $c_{t,i} \leqslant b_i$.

Note that we defined $e_t = [a_t - b_t]_+$. If $a_t \leqslant b_t$, then $e_t = 0$; if $a_t > b_t$, then $e_t = a_t - b_t > 0$: therefore, $e_t \geqslant 0$. We will prove that $\{f_{t,i}\}_{i\in[t]}$ is decreasing by induction such that, for all $i > 1$, we have that $f_{t,i} \geqslant f_{t,i-1} \geqslant 0$. Consider the base case, $i = 2$, we have that $f_{t,1} = e_t \geqslant 0$; we also have, by the inductive hypothesis Assumption 4.9, that $b_1 \geqslant c_{t-1,1}$; or equivalently, $b_1 - c_{t-1,1} \geqslant 0$. Then, Definition 4.7 defines that $e_{t,1} = \min\{f_{t,1}, b_1 - c_{t-1,1}\}$, and since both $f_{t,1}$ and $b_1 - c_{t-1,1}$ are non-negative, we have that $e_{t,1} \geqslant 0$. Therefore, $f_{t,2} = f_{t,1} - e_{t,1} \geqslant f_{t,1}$, that is, $f_{t,2} \geqslant f_{t,1} \geqslant 0$.

Inductively, suppose that, for any $2 < i \leqslant t$, we have that $f_{t,i} \geqslant f_{t,i-1} \geqslant 0$. We will show that $f_{t,i+1} \geqslant f_{t,i} \geqslant 0$. Note that $f_{t,i} \geqslant 0$ is satisfied by the inductive hypothesis already. Then, by Definition 4.7, $e_{t,i} = \min\{f_{t,i}, b_i - c_{t-1,i}\}$. Since, by Assumption 4.9, we have that $b_i \geqslant c_{t-1,i}$, we have that $b_i - c_{t-1,i} \geqslant 0$. Then, since $f_{t,i}$ and $b_i - c_{t-1,i}$ are both non-negative, we have that $e_{t,i}$ is non-negative by definition. Thus, we have that $f_{t,i+1} = f_{t,i} + e_{t,i} \geqslant f_{t,i} \geqslant 0$.

Finally, we need to show that $c_{t,i} \leqslant b_i$. Since $\{f_{t,i}\}$ is a decreasing sequence, as the above inductive proof shows, we have that $e_{t,i} \geqslant 0$ for all $i \in [t-1]$. Then, by

Definition 4.7, for $i < t$, we have that $0 \leqslant e_{t,i} \leqslant b_i - c_{t-1,i}$; or equivalently, $c_{t-1,i} \leqslant b_i$. Definition 4.7 defines $c_{t,t} = a_t - e_t = a_t - [a_t - b_t]_+$: if $a_t \leqslant b_t$, then $[a_t - b_t]_+ = 0$ and we have that $c_{t,t} = a_t \leqslant b_t$; if $a_t > b_t$, then $[a_t - b_t]_+ = a_t - b_t$ and we have that $c_{t,t} = a_t - a_t + b_t = b_t$. Therefore, we conclude that $c_{t,i} \leqslant b_i$ for all $i \leqslant t$. $\square$

Lemma 4.10. *Let $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i) \in \mathbb{Z}^k$ and suppose $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$. Following the notations in Definition 4.7, for all $t \in [k]$, the following holds:*

- $e_t = \sum_{i=1}^{t-1} e_{t,i}$.
- $\sum_{i=1}^{t} a_i = \sum_{i=1}^{t} c_{t,i}$.

Proof: Let $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i) \in \mathbb{Z}^k$ and suppose $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$. We give a proof by induction that, for any $t \in [k]$, $e_t = \sum_{i=1}^{t-1} e_{t,i}$ and $\sum_{i=1}^{t} a_i = \sum_{i=1}^{t} c_{t,i}$.

For the base case, when $t = 1$, Definition 4.7 defines $c_{1,1} = a_1 + e_1$. And since $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$ implies that $a_1 \leqslant b_1$, so $e_1 = [a_1 - b_1]_+ = 0$. Therefore, $a_1 = c_{1,1}$ and e_1, which is not defined, is satisfied trivially.

Inductively, suppose that, for any $t > 1$, we have $e_{t-1} = \sum_{i=1}^{t-2} e_{t-1,i}$ and $\sum_{i=1}^{t-1} a_i = \sum_{i=1}^{t-1} c_{t-1,i}$. We will show that $e_t = \sum_{i=1}^{t-1} e_{t,i}$ and $\sum_{i=1}^{t} a_i = \sum_{i=1}^{t} c_{t-1,i}$. We divide into two cases: when $a_t \leqslant b_t$ and when $a_t > b_t$. Note that, if $a_t \leqslant b_t$, then $e_t = [a_t - b_t]_+ = 0$ and $f_{t,1} = e_t$. From Lemma 4.8, we have that sequence $\{f_{t,1}\}$ is decreasing and $b_i - c_{t,i} \geqslant 0$; therefore, when $a_t \leqslant b_t$, $f_{t,i} = 0$ for all $i \in [t]$ and $e_{t,i} = 0$ for all $i < t$. Thus, $e_t = \sum_{i=1}^{t-1} e_{t,i} = 0$. Suppose that $a_t > b_t$, then by Definition 4.7, $e_t = a_t - b_t$, so $c_{t,t} = b_t$. Notice that Equation (4.2.1) that defines $f_{t,i+1} = f_{t,i} - e_{t,i}$, and rewriting the equation we obtain $f_{t,i} = f_{t,i+1} + e_{t,i}$. Expand $e_t = f_{t,1}$ according to Equation (4.2.1):

$$\begin{aligned} e_t = f_{t,1} &= f_{t,2} + e_{t,1} \\ &= f_{t,3} + e_{t,2} + e_{t,1} \\ &= f_{t,4} + e_{t,3} + e_{t,2} + e_{t,1} \\ &= \dots \\ &= f_{t,t} + e_{t,t-1} + \cdots + e_{t,1}. \end{aligned} \tag{4.2.2}$$

Then, we need to show $f_{t,t} = 0$ so that e_t can be written by the sum of $e_{t,i}$ only.

From Lemma 4.8, we have that $f_{t,t} \geqslant 0$ since sequence $\{f_{t,i}\}_{i \in [t]}$ is non-negative. Suppose that $f_{t,t} > 0$, then we have $e_{t,i} = b_i - c_{t-1,i}$ for all $i \in [t-1]$. That is, we know that $b_i - c_{t-1,i} < f_{t,i}$ for any $i \leqslant t$. This is true since, otherwise, if $e_{t,i}$ is defined by $f_{t,i} \leqslant b_i - c_{t-1,i}$ for some $i < t$, then $f_{t,i+1} = f_{t,i} - e_{t,i} = 0$; then, $f_{t,t} = 0$ since we show that $\{f_{t,i}\}$ is a non-negative and decreasing sequence in Lemma 4.8. Then we obtain the following inequality.

$$a_t - b_t = e_t = f_{t,t} + \sum_{i=1}^{t-1} e_{t,i} > \sum_{i=1}^{t-1} e_{t,i} = \sum_{i=1}^{t-1}(b_i - c_{t-1,i}). \tag{4.2.3}$$

By the inductive hypothesis, we have $\sum_{i=1}^{t-1} a_i = \sum_{i=1}^{t-1} c_{t-1,i}$. Then, substitute and we have that

$$a_t - b_t > \sum_{i=1}^{t-1}(b_i - c_{t-1,i}) = \sum_{i=1}^{t-1} b_i - \sum_{i=1}^{t-1} c_{t-1,i} = \sum_{i=1}^{t-1} b_i - \sum_{i=1}^{t-1} a_i.$$

And move terms in the above equation, we have that

$$\sum_{i=1}^{t} a_i > \sum_{i=1}^{t} b_i$$

However, notice that we assumed $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$, we have $\sum_{i=1}^{t} a_i \leqslant \sum_{i=1}^{t} b_i$, and this is a contradiction. Therefore, $f_{t,t} = 0$, and $e_t = \sum_{i=1}^{t-1} e_{t,i}$.

Finally, we will show $\sum_{i=1}^{t} a_i = \sum_{i=1}^{t} c_{t,i}$. Note that we have $e_t = [a_t - b_t]_+ = \sum_{i=1}^{t-1} e_{t,i}$, $c_{t,t} = a_t - e_t$, and $c_{t,i} = c_{t,i-1} + e_i$. If $a_t \leqslant b_t$, then $e_t = 0$ and $e_{t,i} = 0$ for all $i < t$, so we have $c_{t,t} = a_t$ and $c_{t,i} = c_{t-1,i}$. Combining with the inductive hypothesis, we have that

$$\sum_{i=1}^{t} a_i = a_t + \sum_{i=1}^{t-1} a_i = c_{t,t} + \sum_{i=1}^{t-1} c_{t,i} = \sum_{i=1}^{t} c_{t,i}.$$

Now we suppose that $a_t > b_t$, then $e_t = a_t - b_t$ and $c_{t,t} = b_t$. Thus, we also have the following equation:

$$\begin{aligned}
\sum_{i=1}^{t} c_{t,i} &= c_{t,t} + \sum_{i=1}^{t-1} c_{t,i} \\
&= b_t + \sum_{i=1}^{t-1}(c_{t-1,i} + e_{t,i}) \\
&= b_t + \sum_{i=1}^{t-1} c_{t-1,i} + \sum_{i=1}^{t} e_{t,i} \\
&= b_t + \sum_{i=1}^{t-1} a_i + e_t \\
&= b_t + \sum_{i=1}^{t-1} a_i + a_t - b_t \\
&= \sum_{i=1}^{t} a_i,
\end{aligned} \tag{4.2.4}$$

which was what we want. $\square$

Proof of Theorem 4.5: ($\Longrightarrow$)We first prove the forward direction. Suppose $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}^k$ such that $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$. We will show $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$ for all $\boldsymbol{d} \in \mathbb{D}_k$.

We construct $\boldsymbol{c} \in \mathbb{Z}^k$ use Definition 4.7 and we follow the notations in Definition 4.7. We decompose the proof into two parts such that $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{c}^\top \boldsymbol{d}$ and $\boldsymbol{c}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$, and thus the conclusion follows.

By Lemma 4.8, we have that $c_i \leqslant b_i$, and thus $c_i d_i \leqslant b_i d_i$. Then

$$\boldsymbol{c}^\top \boldsymbol{d} = \sum_{t=1}^{k} c_t d_t \leqslant \sum_{t=1}^{k} b_t d_t = \boldsymbol{b}^\top \boldsymbol{d}.$$

So $\boldsymbol{c}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$ follows immediately.

To prove $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{c}^\top \boldsymbol{d}$, we give an inductive proof such that, for all $t \in [k]$, $\sum_{i=1}^{t} a_{t,i} d_i \leqslant \sum_{i=1}^{t} c_{t,i} d_i$. Let e_t, $\{e_{t,i}\}_{i\in[t]}$, and $\{c_{t,i}\}_{i\in[t]}$ defined as in Definition 4.7, that is, $e_t = [a_t - b_t]_+ = \sum_{i=1}^{t-1} e_{t,i}$ and $c_{t,i} = c_{t-1,i} + e_{t,i}$.

When $t = 1$, we have $c_{1,1} = a_1$ by Lemma 4.10. Then, $a_1 d_1 = c_{1,1} d_1$. This is the base case of the inductive statement.

Inductively, suppose that, for any $t > 1$, we have $\sum_{i=1}^{t-1} a_i d_i \leqslant \sum_{i=1}^{t-1} c_{t-1,i} d_i$. We divide into two cases when $a_t \leqslant b_t$ and when $a_t > b_t$. When $a_t \leqslant b_t$, Lemma 4.10 gives $c_{t,t} = a_t$ and $c_{t,i} = c_{t-1,i}$ for all $i < t$. Then, by the inductive hypothesis,

$$\sum_{i=1}^{t} a_i d_i = a_t d_t + \sum_{i=1}^{t-1} a_i d_i \leqslant c_{t,t} d_t + \sum_{i=1}^{t-1} c_{t-1,i} d_i = c_{t,t} d_t + \sum_{i=1}^{t-1} c_{t,i} d_i = \sum_{i=1}^{t} c_{t,i} d_i,$$

which satisfies the inductive hypothesis.

Consider when $a_t > b_t$, then $e_t = a_t - b_t > 0$ and $c_{t,t} = b_t$. Consider the following sequence of inequalities.

$$\begin{aligned}
\sum_{i=1}^{t} a_i d_i &= a_t d_t + \sum_{i=1}^{t-1} a_i d_t \\
&\leqslant a_t d_t + \sum_{i=1}^{t-1} c_{t-1,i} d_t \quad \text{by the inductive hypothesis} \\
&= a_t d_t + \sum_{i=1}^{t-1} c_{t-1,i} d_t + c_{t,t} d_t - c_{t,t} d_t \qquad (4.2.5) \\
&= (a_t - c_{t,t}) d_t + \sum_{i=1}^{t-1} c_{t-1,i} d_t + c_{t,t} d_t \\
&= e_t d_t + \sum_{i=1}^{t-1} c_{t-1,i} d_t + c_{t,t} d_t
\end{aligned}$$

We write $e_t = \sum_{i=1}^{t-1} e_{t,i-1}$ as in Lemma 4.10; also note that $d_i > d_t$, for all $i < t$, implies that $e_{t,i}d_i \geqslant e_{t,i}d_t$ since $e_{t,i} \geqslant 0$ by Lemma 4.8. Then, we have that

$$\begin{aligned} \sum_{i=1}^{t} a_i d_i &= (\sum_{i=1}^{t-1} e_{t,i})d_t + \sum_{i=1}^{t-1} c_{t-1,i}d_t + c_{t,t}d_t \\ &\leqslant (\sum_{i=1}^{t-1} e_{t,i}d_i) + \sum_{i=1}^{t-1} c_{t-1,i}d_t + c_{t,t}d_t \\ &= \sum_{i=1}^{t-1}(e_{t,i} + c_{t-1,i})d_t + c_{t,t}d_t \\ &= \sum_{i=1}^{t-1} c_{t,i}d_t + c_{t,t}d_t \\ &= \sum_{i=1}^{t} c_{t,i}d_t \end{aligned} \tag{4.2.6}$$

as desired. Therefore, the inductive statement holds for case t.

When $t = k$, by definition $\boldsymbol{c} = (c_i)_{i\in[k]}$ where $c_i = c_{k,i}$, rewrite $\sum_{i=1}^{t} a_i d_i \leqslant \sum_{i=1}^{t} c_i d_t$ as $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{c}^\top \boldsymbol{d}$, which was what we want.

Combine $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{c}^\top \boldsymbol{d}$ and $\boldsymbol{c}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$, and we conclude that $\boldsymbol{a}^\top \boldsymbol{d} \leqslant \boldsymbol{b}^\top \boldsymbol{d}$.

($\Longleftarrow$)Now we prove the converse of the statement. We proceed with a proof by contrapositive. Let $\boldsymbol{a} = (a_i), \boldsymbol{b} = (b_i) \in \mathbb{Z}^k$ and suppose $\boldsymbol{a} \not\preccurlyeq \boldsymbol{b}$. We will show there exists $\boldsymbol{d} \in \mathbb{D}_k$ such that $\boldsymbol{a}^\top \boldsymbol{d} > \boldsymbol{b}^\top \boldsymbol{d}$.

Note that the statement $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$ is defined as, for all $t \in [k]$, we have that $\sum_{i=1}^{t} a_i \leqslant \sum_{i=1}^{t} b_i$; then, the negation of it, $\boldsymbol{a} \not\preccurlyeq \boldsymbol{b}$, is that, exists $t \in [k]$, we have $\sum_{i=1}^{t} a_i > \sum_{i=1}^{t} b_i$. Suppose $t \in [k]$ is the minimal counterexample such that, for all $t' < t$,

$$\sum_{i=0}^{t'} a_i \leqslant \sum_{i=0}^{t'} b_i \tag{4.2.7}$$

and

$$\sum_{i=1}^{t} a_i > \sum_{i=1}^{t} b_i. \tag{4.2.8}$$

Combining the two inequalities, we have

$$a_t - b_t > \sum_{i=1}^{t-1} b_i - \sum_{i=1}^{t-1} a_i \geqslant 0 \tag{4.2.9}$$

Divide both sides by $a_t - b_t$ (which is positive),

$$1 > \frac{\sum_{i=1}^{t-1}(b_i - a_i)}{a_t - b_t}. \tag{4.2.10}$$

Let $d_1 > d_t > 0$ be so that

$$1 > \frac{\sum_{i=1}^{t-1}(b_i - a_i)}{a_t - b_t} \cdot \frac{d_1}{d_t} > \frac{\sum_{i=1}^{t-1}(b_i - a_i)}{a_t - b_t}. \tag{4.2.11}$$

i.e., pick a value for d_t/d_1 in the interval $(\sum_{i=1}^{t-1}(b_i - a_i)/(a_t - b_t), 1)$. Since, for all $\boldsymbol{d} \in \mathbb{D}_k$, $1 < i < t$ implies that $d_1 > d_i$, it follows that

$$\begin{aligned} \frac{\sum_{i=1}^{t-1}(b_i - a_i)}{a_t - b_t} \cdot \frac{d_1}{d_t} &= \sum_{i=1}^{t-1} \frac{(b_i - a_i)d_1}{(a_t - b_t)d_t} \\ &\geqslant \sum_{i=1}^{t-1} \frac{(b_i - a_i)d_i}{(a_t - b_t)d_t} \end{aligned} \tag{4.2.12}$$

Since $a_t > b_t$ and $d_t > 0$, multiplying both sides of Equation (4.2.11) by $(a_t - b_t)d_t$ does not change the direction of the inequality. Combining Equation (4.2.11) and (4.2.12), we have that

$$(a_t - b_t)d_t > \sum_{i=1}^{t-1}(b_i - a_i)d_i.$$

It follows that $\sum_{i=1}^{t} a_i d_i > \sum_{i=1}^{t} b_i d_i$. Let $\epsilon > 0$ be so that $\sum_{i=1}^{t} a_i d_i = \sum_{i=1}^{t} b_i d_i + \epsilon$. Then choose small enough $d_{t+1} > \cdots > d_k$ so that $\sum_{i=t+1}^{k}(b_i - a_i)d_i < \epsilon$ (*i.e.*, use d_i arbitrarily small so that $(b_i - a_i)d_i$ are small). Substitute it in the above equation so that

$$\sum_{i=1}^{t} a_i d_i > \sum_{i=1}^{t} b_i d_i + \sum_{i=t+1}^{k} (b_i - a_i)d_i$$

and so $\sum_{i=1}^{k} a_i d_i > \sum_{i=1}^{k} b_i d_i$ by moving terms, and thus $\boldsymbol{a}^\top \boldsymbol{d} > \boldsymbol{b}^\top \boldsymbol{d}$, which was what we want. □

4.2.1 Modifying the Bound-Generation Algorithm

We now proceed to modify the `Add-Walk-To` program with the partial order $\preccurlyeq$. Notice that, in a complete graph of order n, any path has length at most n since each vertex can appear at most once in the cycle. The order of the vertices in a cycle is not restricted; therefore, consider the bounds in Condition 3.17 induced by upper-bound-cycles, we will need to consider n^k bounds at most. The `Add-Walk-To` algorithm adds all bound-paths to each $\mathfrak{U}_{u,v}$ and $\mathfrak{L}_{u,v}$, which involves $O(n^k)$ elements in each set. With the partial order $\preccurlyeq$, by Theorem 4.5, it suffices to keep only the minimal and maximal elements under $\preccurlyeq$ in `UBW` and `LBW`, respectively, to satisfy Condition 3.17.

We present the modification in Algorithm 3. Instead of using only the bound-paths, we modify the elements to pairs consisting of a bound and a bound-path, We change program `Add-Walk-To` on Line 10, 20, 15 by `Compare-Upper-and-Add-Walk-To`, and change `Add-Walk-To` on Line 12, 22, 17 by `Compare-Lower-and-Add-Walk-To`.

Let S be a set of upper-bound-paths, then we say that *upper-bound-path* $W \in S$ *is a minimal element of* S if $\beta^+(W)$ is a minimal element in all the induced upper bounds of S, $\beta^+(S)$. Similarly, let S be a set of lower-bound-paths, then lower-bound-path $W \in S$ is *a maximal element of* S if $\beta^-(W)$ is a maximal element in all the induced lower bounds of S, $\beta^+(S)$.

Algorithm 3: Bound-Generation-mod

input : Robinson matrix $A \in \mathcal{S}^n[k]$

output: Lookup tables UBW, LBW defined on $i, j \in [n], i < j$. For each $i, j \in [n], i < j$, $\mathtt{UBW}(i,j)$ contains pairs consisting of a bound and an (i,j)-upper-bound-path, where the bound is a minimal element in $\beta^+(\mathfrak{U}_{i,j})$; $\mathtt{LBW}(i,j)$ contains pairs consisting of a bound and an (i,j)-lower-bound-path, where the bound is a maximal element in $\mathfrak{L}_{i,j}$ under $\preccurlyeq$.

for $i \in [n]$ **do**
 for $j = i, \ldots, n$ **do**
  if $a_{i,j} \neq 0$ **then** $\mathtt{UBW}(i,j) \leftarrow \{(\beta^+(\{i,j\}), \langle i,j \rangle)\}$;
  $\mathtt{LBW}(i,j) \leftarrow \{(\beta^-(\{i,j\}), \langle i,j \rangle)\}$;
for $s = 1, \ldots, n$ **do**
 for $i = 1, \ldots, n$ **do**
  for $j = i, \ldots, n$ *and* $i \neq s \neq j$ **do**
   if $i < s < j$ **then**
    foreach $(\boldsymbol{b}_1, W_1) \in \mathtt{UBW}(i,s)$ *and* $(\boldsymbol{b}_2, W_2) \in \mathtt{UBW}(s,j)$ **do**
     Compare-Upper-and-Add-Walk-To$(\mathtt{UBW}(i,j), (\boldsymbol{b}_1 + \boldsymbol{b}_2, W_1 + W_2))$;
    foreach $(\boldsymbol{a}_1, W_1) \in \mathtt{LBW}(i,s)$ *and* $(\boldsymbol{a}_2, W_2) \in \mathtt{LBW}(s,j)$ **do**
     Compare-Lower-and-Add-Walk-To$(\mathtt{LBW}(i,j), (\boldsymbol{a}_1 + \boldsymbol{a}_2, W_1 + W_2))$;
   else if $i < j < s$ **then**
    foreach $(\boldsymbol{b}, W_1) \in \mathtt{UBW}(i,s)$ *and* $(\boldsymbol{a}, W_2) \in \mathtt{LBW}(j,s)$ **do**
     Compare-Upper-and-Add-Walk-To$(\mathtt{UBW}(i,j), (\boldsymbol{b} - \boldsymbol{a}, W_1 + W_2^{\leftarrow}))$;
    foreach $(\boldsymbol{a}, W_1) \in \mathtt{LBW}(i,s)$ *and* $(\boldsymbol{b}, W_2) \in \mathtt{UBW}(j,s)$ **do**
     Compare-Lower-and-Add-Walk-To$(\mathtt{LBW}(i,j), (\boldsymbol{a} - \boldsymbol{b}, W_1 + W_2^{\leftarrow}))$;
   else if $s < i < j$ **then**
    foreach $(\boldsymbol{a}, W_1) \in \mathtt{LBW}(s,i)$ *and* $(\boldsymbol{b}, W_2) \in \mathtt{UBW}(s,j)$ **do**
     Compare-Upper-and-Add-Walk-To$(\mathtt{UBW}(i,j), (\boldsymbol{b} - \boldsymbol{a}, W_1^{\leftarrow} + W_2))$;
    foreach $(\boldsymbol{b}, W_1) \in \mathtt{UBW}(k,i)$ *and* $(\boldsymbol{a}, W_2) \in \mathtt{LBW}(k,j)$ **do**
     Compare-Lower-and-Add-Walk-To$(\mathtt{LBW}(i,j), (\boldsymbol{a} - \boldsymbol{b}, W_1^{\leftarrow} + W_2))$;
return UBW, LBW;

Algorithm 4: Compare-Upper-and-Add-Walk-To

input : A set of pairs consisting of an upper bound and an upper-bound-path S, a pair consisting of an upper bound and an upper-bound-paht $(\boldsymbol{b}, W)$

if W *contains repeating vertices* **then return** ;
foreach $(W', \boldsymbol{b}') \in S$ **do**
    if $\boldsymbol{b}' \preccurlyeq \boldsymbol{b}$ **then return**;
    else if $\boldsymbol{b} \preccurlyeq \boldsymbol{b}'$ **then** $S \leftarrow S \setminus \{(\boldsymbol{b}', W')\}$;
$S \leftarrow S \cup \{(\boldsymbol{b}, W)\}$;

Algorithm 5: Compare-Lower-and-Add-Walk-To

input : A set of pairs consisting of a lower bound and a lower-bound-path S, a pair consisting of a lower bound and a lower-bound-path, $(\boldsymbol{a}, W)$

if W *contains repeating vertices* **then return** ;
foreach $(\boldsymbol{a}', W') \in S$ **do**
    if $\boldsymbol{a} \preccurlyeq \boldsymbol{a}'$ **then return**;
    else if $\boldsymbol{a}' \preccurlyeq \boldsymbol{a}$ **then** $S \leftarrow S \setminus \{(\boldsymbol{a}', W')\}$;
$S \leftarrow S \cup \{(\boldsymbol{a}, W)\}$;

The following two lemmas state that the minimal/maximal bound-paths can be generated from other minimal/maximal bound-paths.

Lemma 4.11. *Let $W \in \mathfrak{U}_{u,v}$, and let s be the largest intermediate vertex in W. Suppose $\beta^+(W)$ is a minimal element in $\beta^+(\mathfrak{U}_{u,v})$, write $W = W_1 + W_2$ with $\langle u, \ldots, s, \ldots, v\rangle$ and $W_1 = \langle u, \ldots, s\rangle, W_2 = \langle s, \ldots, v\rangle$. Then, one of the following condition holds:*

$$\begin{cases} W_1 \text{ is minimal in } \mathfrak{U}^{s-1}_{u,s}, W_2 \text{ is minimal in } \mathfrak{U}^{s-1}_{s,v} & \text{if } u < s < v \quad (4.2.13)\\ W_1 \text{ is minimal in } \mathfrak{U}^{s-1}_{u,s}, W_2^{\leftarrow} \text{ is maximal in } \mathfrak{L}^{s-1}_{v,s} & \text{if } u < v < s \quad (4.2.14)\\ W_1^{\leftarrow} \text{ is maximal in } \mathfrak{L}^{s-1}_{s,u}, W_2 \text{ is minimal in } \mathfrak{U}^{s-1}_{s,v} & \text{if } s < u < v \quad (4.2.15)\end{cases}$$

Proof: Let $W \in \mathfrak{U}_{u,v}$ be such that $\beta^+(W)$ is a minimal element in $\beta^+(\mathfrak{U}_{u,v})$ and let s be the largest intermediate vertex in W. Write $W = W_1 + W_2$ and $W_1 = \langle u, \ldots, s\rangle$ and $W_2 = \langle s, \ldots, v\rangle$. Assume $u < s < v$, we show Equation (4.2.13) by contradiction. Note that $W_1 \in \mathfrak{U}^{s-1}_{u,s}$ and $W_2 \in \mathfrak{U}^{s-1}_{s,v}$.

Toward contradiction, suppose $W \in \mathfrak{U}^{s}_{u,v}$ is minimal, $W_1 \in \mathfrak{U}^{s-1}_{u,s}$ is not minimal, and $W_2 \in \mathfrak{U}^{s-1}_{s,v}$ is minimal. Since W_1 is not minimal, there exists some W_1' such that $\beta^+(W_1') \preccurlyeq \beta^+(W_1)$. By Theorem 4.5, for any $\boldsymbol{d} \in \mathbb{D}_k$, $\beta^+(W_1')^\top \boldsymbol{d} \leqslant \beta^+(W_1)^\top \boldsymbol{d}$, and thus

$$\beta^+(W_1')^\top \boldsymbol{d} + \beta^+(W_2)^\top \boldsymbol{d} \leqslant \beta^+(W_1)^\top \boldsymbol{d} + \beta^+(W_2)^\top \boldsymbol{d} = \beta^+(W)^\top \boldsymbol{d}.$$

Convert back to the partial order, $\beta^+(W_1' + W_2) \preccurlyeq \beta^+(W_1 + W_2) = \beta^+(W)$, which implies W is not minimal in $\mathfrak{U}^{s}_{u,v}$, which contradicts to the assumption. The same case happens when W_2 is not minimal in $\mathfrak{U}_{s,v}$, or both W_1 not minimal in $\mathfrak{U}^{s-1}_{u,s}$ and W_2 is not minimal in $\mathfrak{U}^{s-1}_{s,v}$.

Similarly, the same logic applies to Equation (4.2.14), Equation (4.2.15). □

By symmetry, the similar statement applies to lower-bound-paths.

Lemma 4.12. *Let $W \in \mathfrak{L}^{s}_{u,v}$ and $W = \langle u, \ldots, s, \ldots, v\rangle$. Write $W = W_1 + W_2$ with $W_1 = \langle u, \ldots, s\rangle, W_2 = \langle s, \ldots, v\rangle$. Then*

$$\begin{cases} W_1 \text{ is maximal in } \mathfrak{L}^{s-1}_{u,s}, W_2 \text{ is maximal in } \mathfrak{L}^{s-1}_{s,v} & \text{if } u < s < v \quad (4.2.16)\\ W_1 \text{ is maximal in } \mathfrak{L}^{s-1}_{u,s}, W_2^{\leftarrow} \text{ is minimal in } \mathfrak{U}^{s-1}_{v,s} & \text{if } u < v < s \quad (4.2.17)\\ W_1^{\leftarrow} \text{ is minimal in } \mathfrak{U}^{s-1}_{s,u}, W_2 \text{ is maximal in } \mathfrak{L}^{s-1}_{s,v} & \text{if } s < u < v \quad (4.2.18)\end{cases}$$

These two lemmas apply to `Bound-Generation-mod`, such that all the minimal upper-bound-paths and maximal lower-bound-paths are enumerated when `UBW`(i, j) stores only the minimal elements and `LBW`(i, j) stores only the maximal elements at each iteration.

4.3 Finding a Threshold Vector

Given Robinson matrix $A \in \mathcal{S}^n[k]$, recall that the iterative procedure given in Definition 3.22 involves a threshold vector $\boldsymbol{d} \in \mathbb{D}_k$ that satisfies Condition 3.17. We devote this chapter on how to obtain one of such vector $\boldsymbol{d}$.

In Condition 3.17, the system of inequalities is described in terms of the induced bound of lower- and upper-bound-walks, $\boldsymbol{a}, \boldsymbol{b}$, so that $\boldsymbol{a}^\top \boldsymbol{d} < \boldsymbol{b}^\top \boldsymbol{d}$. Notice, it is equivalent to state Condition 3.17 as, for all $u, v \in [n]$, for all $\boldsymbol{a} = \beta^+(W_2) \in \beta^-(\mathfrak{L}_{u,v}), \boldsymbol{b} = \beta^+(W_1) \in \beta^+(\mathfrak{U}_{u,v})$ $0 < (\boldsymbol{b} - \boldsymbol{a})^\top \boldsymbol{d}$. Then, consider it in terms of bound-walks, W_1 is a (u, v)-upper-bound-walk and W_2 is a (u, v)-lower-bound-walk, and thus, $W_1 + W_2^{\leftarrow}$ is an upper-bound-walk with the vertices on the two ends are the same (It may not be a cycle since there may be common vertices in W_1, W_2). Recall Lemma 3.19 and Lemma 3.21, we shown that by "detaching" bound-cycles from a bound-walk, it results in a "better" bound-walk in Condition 3.17. Therefore, let $\mathcal{C}$ be the set of upper-bound-cycles, and rewrite such that $\boldsymbol{d}$ satisfies Condition 3.17:

$$\text{For all } C \in \mathcal{C}, \quad \beta^+(C)^\top \boldsymbol{d} > 0. \tag{4.3.1}$$

Any cycle in a connected graph with vertex set $[n]$ has length at most n. Any edge in the cycle can contribute at most 1 to the coefficients of the induced bound. Thus we have that, for any upper-bound-cycle C,

$$\beta^+(C) \in \mathbb{Z}_n^k := \left\{ \boldsymbol{a} \in \mathbb{Z}^k : \sum_{i=1}^{k} |a_i| \leqslant n \right\}.$$

In particular, no coefficient of a bound, induced by a upper-bound-path, can have absolute value more than n. This implies that the number of upper-bound-cycles is at most n^k, and thus the number of inequalities in (4.3.1) is at most n^k.

Let variable $z \in \mathbb{R}$, and $\mathbf{1}$ be the all-one vector. Define the linear program that finds a solution to $\boldsymbol{d} = (d_i)$, where the objective function is z. Each row of the constraint matrix, $\boldsymbol{B}$, is $\boldsymbol{b}^\top$ for each $\boldsymbol{b} \in \beta^+(\mathcal{C})$.

$$\begin{aligned} \text{Maximize} \quad & z \\ \text{subject to} \quad & \boldsymbol{B}\boldsymbol{d} \geqslant z\mathbf{1} \\ & d_i - d_{i+1} \geqslant z \quad \forall i \in [k-1] \\ \text{and} \quad & d_k \geqslant z \end{aligned} \tag{4.3.2}$$

Notice, $\boldsymbol{d} \in \mathbb{D}_k$ has no solution if the solution of z being $z \leqslant 0$. Precisely, each entry of $\boldsymbol{B}\boldsymbol{d} \geqslant z\mathbf{1}$ corresponds to $\boldsymbol{b}^\top \boldsymbol{d} > 0$ as in (4.3.1); each $d_i - d_{i+1} \geqslant z$ corresponds to

$\boldsymbol{d} = (d_i)$ is a strictly decreasing vector; $d_k \geqslant z$ corresponds to $d_k > 0$. If the maximum solution of z is non-positive, it implies that, for all solutions of $\boldsymbol{d} \in \mathbb{R}^k$, either $\boldsymbol{d} \notin \mathbb{D}_k$ or there is $\boldsymbol{b} \in \beta^+(\mathcal{C})$ such that $\boldsymbol{b}^\top \boldsymbol{d} \leqslant 0$.

4.4 Time Complexity of Computing a Uniform Embedding

In this section, we discuss the computation of a uniform embedding of a Robinson matrix A. Consider the following steps.

1. Generate all upper- and lower-bound-paths using the `Bound-Generation-mod` algorithm (Algorithm 3) and compute all upper-bound-cycles $\mathcal{C}$.

2. Use a linear program solver to solve linear program (4.3.2). If linear program (4.3.2) does not have a solution, then EXIT and print "No Solution"; otherwise, return a solution $\boldsymbol{d}$.

3. Apply $\beta^+(\mathfrak{U}_{u,v}), \beta^-(\mathfrak{L}_{u,v})$ to compute a uniform embedding Π with respect to $\boldsymbol{d}$ using the formula in Definition 3.22. **Exit** and return Π.

Recall, the complexity of the Floyd-Warshall algorithm is dominated by computing all s-(i,j)-paths, such that it enumerates on all $i, j, s \in [n]$; thus, the complexity of the algorithm is $O(n^3)$ where n is the number of vertices. Particularly, there is a comparison on $D(i,j)$ and $D(i,s) + D(s,j)$; but $D(i,j)$ are all real numbers, the cost is a constant.

In `Bound-Generation-mod` algorithm, the same enumeration is needed. Then, instead of the comparison on the real numbers, the algorithm needs to enumerate, for each $u, s, v \in [n]$, all the bound-paths in a pair of bound-paths on Line 10, 20, or 15, and on Line 12, 22, or 17; then use `Compare-Upper-and-Add-Walk-To` or `Compare-Lower-and-Add-Walk-To` to add a path to the corresponding set. Notice, without using $\preccurlyeq$, there are $O(n^k)$ bounds exists in each of $\mathfrak{U}_{u,v}$ and $\mathfrak{L}_{u,v}$. By apply $\preccurlyeq$ to the algorithm, we may reduce the size of bounds in each set by taking the minimal/maximal elements, we denote this size by $O(M)$. Then, the complexities of `Compare-Upper-and-Add-Walk-To` and `Compare-Lower-and-Add-Walk-To` are both $O(nM)$ since they need to verify that the given upper-bound-walk is indeed an upper-bound-cycle and compare it to all elements in the corresponding sets. On each of Line 10, 20, or 15, and on Line 12, 22, or 17, it enumerates all the upper- and lower-boundpaths, with complexity $O(M^2)$. Therefore, we conclude that the complexity of `Bound-Generation-mod` with `Compare-Upper-and-Add-Walk-To` and

`Compare-Lower-and-Add-Walk-To` is $O(n^3M^2(nM)) = O(n^4M^3)$. Unfortunately, we could not determine the exact relation between $O(M)$ and $O(n^k)$.

Use the linear program solver in [26] by Vaidya, the cost to solve a linear program is $O((n+d)^{1.5}nL)$ where d is the number of constraints and L is the number of bits of the entry values. In this thesis, we focus on the combinatorial properties of the algorithm; therefore, we assume the bound on bit size of each entries, L, to be a constant since it is more relevant to the implementation of the algorithm. In our inequality system 4.3.2, the number of constraints d is M. Thus, the complexity of solving linear program 4.3.2 is $O((n+M)^{1.5}n)$.

Finally, for the calculation of Π, Definition 3.22, for each $v \in [n]$, it enumerates $O(M)$ elements in $\beta^+(S)$ in each set $\beta^+(\mathfrak{U}_{i,v})$ or $\beta^-(\mathfrak{L}_{i,v})$ for $i \in [v]$. Then the complexity is $O(n^2M)$.

So we conclude the overall complexity of the entire procedure is $O(n^4M^3 + (n+M)^{1.5}n + n^2M) = O(n^4M^3)$, if we assume L is a constant.

4.5 The Uniform Embedding Algorithm of Case of $k = 2$

In this section, we consider the problem of finding the uniform embedding when $k = 2$, *i.e.*, the given matrix $A = (a_{i,j})$ with $a_{i,j} \in \{0, 1, 2\}$, and $\boldsymbol{d} = (d_1, d_2)^\top$. Recall that the size of each entry of $\texttt{UBW}, \texttt{LBW}$ is $O(M)$, which cannot be determined precisely. However, in the case of $k = 2$, we can determine the upper bound on the sizes in terms of n. Also, we propose a combinatorial procedure to compute the uniform embedding, if the matrix has a uniform embedding, *i.e.*, without using the linear program as in Section 4.3.

We rewrite the constraints in linear program 4.3.2 when $k = 2$: Let $A \in \mathcal{S}^n[2]$ and let $\boldsymbol{B}$ be constraint matrix. Notice that $\boldsymbol{B}$ is a matrix of size $M \times 2$, where M is the number of minimal elements in $\beta^+(\mathcal{C})$. Each row of $\boldsymbol{B}$ being $\boldsymbol{b}^\top$, where $\boldsymbol{b} = (b_1, b_2)^\top$ are the minimal elements in $\beta^+(\mathcal{C})$. Then, rewrite linear program as the following, for $z \in \mathbb{R}$ and find solution for $\boldsymbol{d} = (d_1, d_2)^\top$,

$$
\begin{array}{rll}
\text{Maximize} & z & \\
\text{subject to} & b_1d_1 + b_2d_2 \geqslant z & \text{for } \boldsymbol{b} = (b_1, b_2)^\top \in \beta^+(\mathcal{C}) \\
& d_1 - d_2 \geqslant z & \\
\text{and} & d_2 \geqslant z &
\end{array}
$$

Now we rewrite the constraints, $\boldsymbol{Bd}$, without z. For each $\boldsymbol{b} = (b_1, b_2)^\top$,

$$\begin{aligned} b_1 d_1 + b_2 d_2 \geqslant z &\iff \begin{cases} d_2 > -b_1 d_1/b_2 & \text{if } b_2 > 0 \\ d_2 < -b_1 d_1/b_2 & \text{if } b_2 < 0 \end{cases} \\ d_1 - d_2 \geqslant z &\iff d_2 < d_1. \end{aligned} \tag{4.5.1}$$

Notice that these constraints restrict a region in $\mathbb{R}^2$. Also notice that the boundaries of these restrictions are linear equations that pass the origin; and since the solutions of $\boldsymbol{d}$ is restricted to $d_1 > d_2 > 0$. The constraints can be linearly ordered: Let $\boldsymbol{b}_1 = (b_1, b_2)^\top, \boldsymbol{b}_2 = (b_3, b_4)^\top$ be two bounds in the system, then $-b_1 d_2/b_2 > -b_3 d_2/b_4$ if and only if $-b_1/b_2 > -b_3/b_4$. Therefore, we can fix $d_1 = 1$ without loss of generality, and rewrite each constraint $\boldsymbol{b} = (b_1, b_2)^\top$ as the following form:

$$\begin{aligned} d_2 > -b_1/b_2 \quad \text{if } b_2 > 0, \\ d_2 < -b_1/b_2 \quad \text{if } b_2 < 0. \end{aligned}$$

while $d_2 < 1$. Let $R = \min\{1\} \cup \{-b_1/b_2 : b_2 < 0\}$ and $L = \max\{0\} \cup \{-b_1/b_2 : b_2 > 0\}$ and notice $\boldsymbol{d}$ has a solution if and only if $L < R$. Recall that there are finitely many upper-bound-cycles, so the minimum and the maximum belong to elements in the two sets that define L and R. Therefore, if $L < R$, then $(1, d_2)^\top$ is a solution of $\boldsymbol{d}$ for any $d_2 \in (L, R)$. Otherwise, if $L > R$, then $\boldsymbol{d}$ does not have a solution.

4.5.1 The Size of Bounds when $k = 2$

In this subsection, we discuss the number of minimal elements in $\beta^+(\mathcal{C})$, M, under the partial order $\preccurlyeq$, produced by `Bound-Generation-mod`.

Lemma 4.13. *Let $A \in \mathcal{S}^n[2]$, i.e., if $k = 2$, and let $\mathcal{C}$ be the set of all upper-bound-cycles. Then the number of minimal element in $\beta^+(\mathcal{C})$ under $\preccurlyeq$ is at most $2n$.*

First consider a partition of the set $S = \{\boldsymbol{b} = (b_1, b_2)^\top : \boldsymbol{b} \in \mathbb{Z}_k^2, b_1 + b_2 \leqslant n\}$ into n cases, for $t \in [n]$, $S_t = \{(b_1, b_2)^\top \in \mathbb{Z}^2 : |b_1| + |b_2| = t\}$. For each t, partition $S_t = S_t^1 \cup S_t^2$ where

$$\begin{aligned} S_t^1 &= \{(-t+i, i)^\top : 0 \leqslant i \leqslant t\} \cup \{(i, t-i)^\top : 0 \leqslant i \leqslant t\}, \text{ and} \\ S_t^2 &= \{(-t+i, -i)^\top : 1 \leqslant i \leqslant t\} \cup \{(i, -t+i)^\top : 0 \leqslant i \leqslant t\}. \end{aligned}$$

Notice that each S_t^j forms a chain.

1. First consider subset $\{(-t+i_1, i_1)^\top : 0 \leqslant i_1 \leqslant t\}$ of S_t^1, $(-t+i_1, i_1)^\top \preccurlyeq (-t+i_1+1, i_1+1)^\top$ for each $0 \leqslant i_1 < t$. In the subset $\{(i_2, t-i_2)^\top : 1 \leqslant i_2 \leqslant t\}$, $(i_2, t-i_2)^\top \preccurlyeq (i_2+1, t-i_2-1)^\top$ for $1 \leqslant i_2 < t$. When $i_1 = t$ and $i_2 = 0$, the elements in the two subsets are both $(0,t)^\top$. Therefore, S_t^1 forms a chain.

2. In subset $\{(-t+i_1, -i_1)^\top : 1 \leqslant i_1 \leqslant t\}$ of S_t^2, $(-t+i_1, -i_1)^\top \preccurlyeq (-t+i_1+1, -i_1-1)^\top$ for each $1 \leqslant i_1 < t$. In the subset $\{(i_2, -t+i_2)^\top : 0 \leqslant i_2 \leqslant t\}$, $(i_2, -t+i_2)^\top \preccurlyeq (i_2+1, -t+i_2+1)^\top$ for $0 \leqslant i_2 < t$. When $i_1 = t$ and $i_2 = 0$, both elements are $(0, -t)^\top$. Therefore, S_t^2 forms a chain.

Recall that there are only finitely many upper-bound-cycles in $\mathcal{C}$, shown in Lemma 3.21. Apply Dilworth's theorem, such that, set S is a finite partially ordered set that can be partitioned into $2n$ chains, and thus, the number of minimal elements is bounded by $2n$.

Example 4.14. Consider any example on Robinson matrix of order 5, $\mathcal{S}^n[2]$. Then, all the bounds with the absolute value of the coefficients summing up to 5 are in form of:

$$\begin{bmatrix}-5\\0\end{bmatrix} \preccurlyeq \begin{bmatrix}-4\\1\end{bmatrix} \preccurlyeq \begin{bmatrix}-3\\2\end{bmatrix} \cdots \preccurlyeq \begin{bmatrix}2\\3\end{bmatrix} \preccurlyeq \begin{bmatrix}3\\2\end{bmatrix} \preccurlyeq \begin{bmatrix}4\\1\end{bmatrix}$$

$$\begin{bmatrix}-4\\-1\end{bmatrix} \preccurlyeq \begin{bmatrix}-3\\-2\end{bmatrix} \cdots \preccurlyeq \begin{bmatrix}2\\-3\end{bmatrix} \preccurlyeq \begin{bmatrix}3\\-2\end{bmatrix} \preccurlyeq \begin{bmatrix}4\\-1\end{bmatrix}$$

4.5.2 The Combinatorial Procedure and its Complexity

We summarize and present the combinatorial procedure to compute a uniform embedding of a Robinson matrix when $k = 2$. Given Robinson matrix $A \in \mathcal{S}^n[k]$,

1. Generate all upper- and lower-bound-paths using the `Bound-Generation-mod` algorithm (Algorithm 3). Compute all pairs of upper bound and upper-bound-cycle, $(\beta^+(C), C)$, while keeping the pairs only if $\beta^+(C)$ is minimal in $\beta^+(\mathcal{C})$.

2. Fix $d_1 = 1$. For each bound $\beta^+(C)$, convert it into a lower or upper bound on d_2, as in (4.5.1). Only keep two cycles $(\boldsymbol{b}_1, \beta^+(C_1))$ and $(\boldsymbol{b}_2, \beta^+(C_2))$ such that, $\boldsymbol{b}_1 = (b_1, b_2)^\top$ has $b_2 < 0$ and produces the smallest $-b_1/b_2$, $\boldsymbol{b}_2 = (b_3, b_4)^\top$ has $b_4 > 0$ and produces the largest $-b_3/b_4$. Notice that $\boldsymbol{b}_1$ produces L and $\boldsymbol{b}_2$ produces R. If $L \geqslant R$, then print NO SOLUTION. **Exit** and return C_1 and C_2. Otherwise, $L < R$, then choose d_1, d_2 so that d_2/d_1 lies in (L, R).

3. Compute a uniform embedding Π with respect to (d_1, d_2) using the formula given in Definition 3.22. **Exit** and return Π.

The `Bound-Generation-mod` algorithm still enumerates all $i, j, s \in [n]$ in a triply nested loop. In each iteration, Line 10, 11, 14, 16, 19, or 21 enumerates at most $(2n)^2$ bound-paths since there are at most $2n$ minimal or maximal elements in each entry of `UBW` and `LBW`. On each Line 11, 12, 15, 17, 20, or 22, it checks does the generated bound-walk is actually a bound-path and compares $2n$ elements to check its minimality, which costs $O(n^2)$. Therefore, the complexity of `Bound-Generation-mod` is $O(n^3 \cdot n^2 \cdot (n^2)) = O(n^7)$. Step 2 costs $O(n)$ since $\mathcal{C}$ has at most $2n$ elements; Step 3 costs $O(n^2 \cdot n) = O(n^3)$ steps to compute a uniform embedding using Definition 3.22, based on the output of `Bound-Generation-mod`. Therefore, the overall complexity of computing a uniform embedding of matrices in $\mathcal{S}^n[2]$ is $O(n^7)$. Note that the input is a Robinson matrix and it consists of $\Theta(n^2)$ entries. Let $N = n^2$, the complexity of the combinatorial algorithm for $k = 2$, with respect to the input size N, is $O(N^{3.5})$.

Notice, this analysis only applies when $k = 2$. For any $k > 2$, we could not apply Dilworth's theorem to the bounds (k-dimensional integer vectors) to form chains; we also could not fix $d_1 = 1$ so that determine the restrictions on the rest of d_i, for all $i > 1$, since there is no linear order on d_i's. This explains why this thesis ends with the case of $k = 2$, but not any higher case.

Chapter 5

Conclusion and Future Works

In this thesis, we studied how to identify whether a Robinson matrix has a uniform embedding or not, and we provide a complete procedure to compute the uniform embedding if there exists one. We first established a sufficient and necessary condition on a Robinson matrix to have a uniform embedding, assuming the threshold vector and all the implied bounds from the original inequalities are given. The necessity is immediate. We proved the sufficiency by constructing a uniform embedding with a threshold vector and the implied bounds. We generalized the bounds by k-dimensional vectors, and we employed a Floyd-Warshall-like algorithm, `Bound-Generation`, to generate all the implied bounds. We write the solution of the threshold vector in a system of inequalities. If the system has a solution then there exists a uniform embedding.

We also proposed a partial order on bounds (or k-dimensional vectors) that associates with threshold vectors. With this partial order, we reduce the complexity of `Bound-Generation` algorithm.

Finally, we provide an $O(N^{3.5})$ combinatorial algorithm of computing the uniform embedding when the given Robinson matrix has entries that are either 0, 1, or 2.

Bibliography

[1] Jonathan E Atkins, Erik G Boman, and Bruce Hendrickson. A spectral algorithm for seriation and the consecutive ones problem. *SIAM Journal on Computing*, 28(1):297–310, 1998.

[2] Kenneth P. Bogart and Douglas B. West. A short proof that 'proper = unit'. *Discrete Mathematics*, 201(1):21–23, 1999.

[3] Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976.

[4] Huda Chuangpishit, Mahya Ghandehari, Matthew Hurshman, Jeannette Janssen, and Nauzer Kalyaniwalla. Linear embeddings of graphs and graph limits. *Journal of Combinatorial Theory, Series B*, 113:162–184, 2015.

[5] Huda Chuangpishit, Mahya Ghandehari, and Jeannette Janssen. Uniform linear embeddings of graphons. *European Journal of Combinatorics*, 61:47–68, 2017.

[6] Derek G Corneil. A simple 3-sweep lbfs algorithm for the recognition of unit interval graphs. *Discrete Applied Mathematics*, 138(3):371–379, 2004.

[7] Derek G Corneil, Hiryoung Kim, Sridhar Natarajan, Stephan Olariu, and Alan P Sprague. Simple linear time recognition of unit interval graphs. *Information processing letters*, 55(2):99–104, 1995.

[8] James Edward Doran, Jim Doran, Frank E Hodson, and Frank Roy Hodson. *Mathematics and computers in archaeology.* Harvard University Press, 1975.

[9] Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, June 1962.

[10] D. R. Fulkerson and O. A. Gross. Incidence matrices with the consecutive 1's property. *Bulletin of the American Mathematical Society*, 70(5):681 – 684, 1964.

[11] Frédéric Gardi. The Roberts characterization of proper and unit interval graphs. *Discrete Mathematics*, 307(22):2906 – 2908, 2007.

[12] Michel Habib, Ross McConnell, Christophe Paul, and Laurent Viennot. Lex-bfs and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234(1-2):59–84, 2000.

[13] David Kendall. Incidence matrices, interval graphs and seriation in archeology. *Pacific Journal of mathematics*, 28(3):565–570, 1969.

[14] M. Laurent and M. Seminaroti. A lex-bfs-based recognition algorithm for robinsonian matrices. *Discrete Applied Mathematics*, 222:151 – 165, 2017.

[15] Monique Laurent and Matteo Seminaroti. Similarity-first search: a new algorithm with application to robinsonian matrix recognition. *SIAM Journal on Discrete Mathematics*, 31(3):1765–1800, 2017.

[16] Innar Liiv. Seriation and matrix reordering methods: An historical overview. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 3(2):70–91, 2010.

[17] Peter J Looges and Stephan Olariu. Optimal greedy algorithms for indifference graphs. *Computers & Mathematics with Applications*, 25(7):15–25, 1993.

[18] Boris G Mirkin and Sergej N Rodin. *Graphs and Genes*. Biomathematics. Springer, 1984.

[19] W. M. Flinders Petrie. Sequences in prehistoric remains. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 29(3/4):295–301, 1899.

[20] Pascal Préa and Dominique Fortin. An optimal algorithm to recognize Robinsonian dissimilarities. *Journal of Classification*, 31(3):351–385, 2014.

[21] Fred S Roberts. *Indifference graphs*, pages 139–146. Academic Press, 1969.

[22] Fred S Roberts. On the compatibility between a graph and a simple order. *Journal of Combinatorial Theory, Series B*, 11(1):28–38, 1971.

[23] W. S. Robinson. A method for chronologically ordering archaeological deposits. *American Antiquity*, 16(4):293–301, 1951.

[24] Donald J Rose, R Endre Tarjan, and George S Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on computing*, 5(2):266–283, 1976.

[25] Matteo Seminaroti et al. *Combinatorial algorithms for the seriation problem*. PhD thesis, "Tilburg University", 2016.

[26] P. M. Vaidya. Speeding-up linear programming using fast matrix multiplication. In *30th Annual Symposium on Foundations of Computer Science*, pages 332–337, 1989.